COSC 522 Final Project Group 5: Fraudulent Transaction Detection

NOAH ARCHER, The University of Tennessee, Knoxville, USA
TYLER HERNDON, The University of Tennessee, Knoxville, USA
COREY LOVETTE, The University of Tennessee, Knoxville, USA
STEPHEN WILLIAMS, The University of Tennessee, Knoxville, USA

In this project, the goal was to take a set of inputs regarding some transaction and be able to detect whether that transaction is fraudulent or not. The motivation for this type of project was to be able to prevent fraudulent transactions from processing which protects the account user who would have been a victim if the transaction was not flagged as fraudulent and prevented. To train a machine learning model to detect fraudulent transactions, a dataset from Kaggle, aptly named "Credit Card Fraud", was used, and contains inputs such as transaction type, amount of transaction, transaction origination and destination, names of originator and receiver, and before and after balances of each account. Another goal of this project was to compare two types of machine learning solutions to complete this task. The two models chosen for this project were a deep neutral network binary classification model and a random forest model. The results of these two models were compared to find the best model in hope to help prevent fraudulent transactions from taking place.

CCS Concepts: • **Computing methodologies** → *Artificial intelligence*; **Classification and regression trees**; **Neural networks**.

Additional Key Words and Phrases: Machine, Deep, Learning, Neural, Network, AI, Fraudulent, Transaction, Detection

ACM Reference Format:
Noah Archer, Tyler Herndon, Corey Lovette, and Stephen Williams. 2024. COSC 522 Final Project Group 5: Fraudulent Transaction Detection. 1, 1 (May 2024), 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Credit cards have become a staple in modern day society. With the promises to build credit, earn rewards for spending money, and access to quick cash individuals may or may not can afford, most people in today's society have at least one credit card, but many even have multiple credit cards. As a result, there are nearly two billion credit card transactions that occur every day globally, and one of the biggest concerns that consumers have is credit card fraud [4][8]. Because of the large volume of transactions that occur in just the span of a single day and the sheer amount of data associated with each transaction, it is difficult to visually scan and identify fraudulent transactions at random. However, these characteristics are what set the stage for a machine learning application that has potential to put consumers at peace of mind when it comes to the security of their credit card transactions.

Authors' Contact Information: Noah Archer, The University of Tennessee, Knoxville, Knoxville, Tennessee, USA, narcher1@vols.utk.edu; Tyler Herndon, The University of Tennessee, Knoxville, Knoxville, Tennessee, USA, nherndo1@vols.utk.edu; Corey Lovette, The University of Tennessee, Knoxville, Knoxville, Tennessee, USA, clovett3@vols.utk.edu; Stephen Williams, The University of Tennessee, Knoxville, Knoxville, Tennessee, USA, swill200@utk.edu.

Manuscript submitted to ACM

Machine learning algorithms excel at detecting statistical patterns within large amounts of data that can remain undetectable to the human eye. For this particular application, the goal is to take in a set of inputs regarding some credit card transaction and decide whether or not that transaction is legitimate and therefore acceptable, or if the transaction is fraudulent and therefore should be rejected. This type of machine learning application, where there is only two possible outputs of the algorithm, is referred to as binary classification.

One of the difficulties with detecting fraudulent transactions is that less than one percent of all transactions are fraudulent [3]. This means that data is heavily imbalanced, and a prediction model could be right over 99% of the time by predicting the transaction as legitimate however this provides no protection to the consumer. On the other hand, if a model is generated that predicts too many fraudulent transactions that are actually legitimate transactions performed by the consumer, this can be even more frustrating from the consumer's perspective since their legitimate transactions could be denied. One possible solution for this issue is to allow the consumers phone to act as a secondary confirmation for transactions that are flagged as fraudulent. When a possible fraudulent transaction is detected, an alert can be sent to the user's phone to override the fraudulent flagging. However, if consumers are consistently having to utilize their phone due to too many alarms, this will be equally frustrating from their perspective. With these notes in mind, the goal of this project is to create an efficient machine learning model that can quickly identify fraudulent transactions and flag them to help protect consumers and prevent fraud. The target for model accuracy will be 95% overall while prioritizing on minimizing undetected fraudulent transactions to put consumer security first.

2 RELATED WORK

There has been lots of research into credit card fraud detection and detection systems. Over the years several fraud detection techniques have come about such as neural networks, data mining, distributed data mining, and many more.

Research on credit card fraud uses both Machine Learning and Deep Learning algorithms, both of which we have used in our own research here. When working with credit card fraudulent data, there are two points to consider: (1) the methods that are available to us today through existing research and frameworks/packages and (2) how to handle the imbalanced data [7].

In 2019, Yashvi Jain, NamrataTiwari, and Shripriya Dubey researched various techniques for detecting fraudulent credit card transactions [1]. They researched using techniques such as support vector machines, decision trees, K-Nearest Neighbours, and a few others. In their paper, they observed that decision trees, K-Nearest Neighbours and support vector machines gave a medium level accuracy while other techniques gave a high detection rate, such as Neural Networks, naive Bayes, and fuzzy systems. The two algorithms that were specifically pointed out that performed well in all parameters were Naive Bayesian Networks and K-Nearest Neighbours. However, these algorithms are very expensive to train.

In 2019, Sahayasakila V, D.Kavya Monisha, Aishwarya, and Sikhakolli Venkatavisalakshiswshai Yasaswi performed research in trying to solve the problem of imbalances data [6]. During their research, they mainly aimed to improve the convergence speed and to solve the data imbalance problem with two techniques they call Whale Optimization Techniques (WOA) and SMOTE (Synthetic Minority Oversampling Techniques). Combining these two techniques into one algorithm, they were able to improve the convergence speed, reliability, and efficiency,

In 2018, Navanushu Khare and Saad Yunus Sait researched into decision trees, random forest, support vector machines, and logistic regression [2]. While working with each algorithm, they used highly skewed datasets. They found that the accuracy of the logistic regression was 97.7% while decision trees had an accuracy of 95.5%, 98.6% for random forest,

Feature Number	Feature Name	Description
1	distance_from_home	Distance from home where transaction occurred
2	distance_from_last_transaction	Distance from where last transaction occurred
3	ratio_to_median_purchase_price	Ratio of transaction price to median purchase price
4	repeat_retailer	True if user has purchased from retailer prior
5	used_chip	True if transaction completed through chip (credit card)
6	used_pin_number	True if transaction completed through PIN number
7	online_order	True if transaction completed in online order
8	fraud	True if transaction is fraudulent

Table 1. Features contained in "Credit Card Fraud"

Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Feature 8
57.88	0.31	1.95	1	1	0	0	0
10.83	0.18	1.29	1	0	0	0	0
5.09	0.81	0.43	1	0	0	1	0
2.25	5.60	0.36	1	1	0	1	0
44.19	0.57	2.22	1	1	0	1	0

Table 2. First five rows from dataset rounded to two decimal places (feature number explanation can be found in Table 1)

and 97.5% for support vector machines. They concluded that the random forest algorithm had the highest accuracy amongst the rest of the other algorithms and said to be the best algorithm to detect credit card fraudulent transactions.

3 DATA

All of the data utilized in this project came from the "Credit Card Fraud" dataset posted on *Kaggle*, a popular data science community website where lots of various datasets exist for various machine learning opportunities, by Dhanush Narayanan R [5]. This dataset was chosen for its high reviews, features that are useful and easy to work with, and overall size that was determined to be appropriate for the scale of this project. The dataset is described by the author as being sourced by an unnamed institute which is likely due to the desired protection of financial records.

There were one-million entries contained in the data with no duplicate or partially missing records. Each entry contains numeric data for 8 different columns: distance_from_home, distance_from_last_transaction, ratio_to_median_purchase_price, repeat_retailer, used_chip, used_pin_number, online_order, and fraud. The description of each of the features is given in Table 1 based on the description given by the author in the dataset description, and the first five rows of the dataset can be found in Table 2. One tool that is useful for analyzing features and how they can relate to one another in a large dataset is a correlation matrix. A correlation matrix shows the correlation coefficient between each pair of features, and the calculated correlation matrix for this dataset can be found in Figure 1. Of the one-million total entries, only 87,403 of the recorded transactions are flagged as being fraudulent which is just under 9%. This will lend itself to some difficulty later on during model training in regards to biasing, but countermeasures to prevent biasing will be discussed in the methods section.

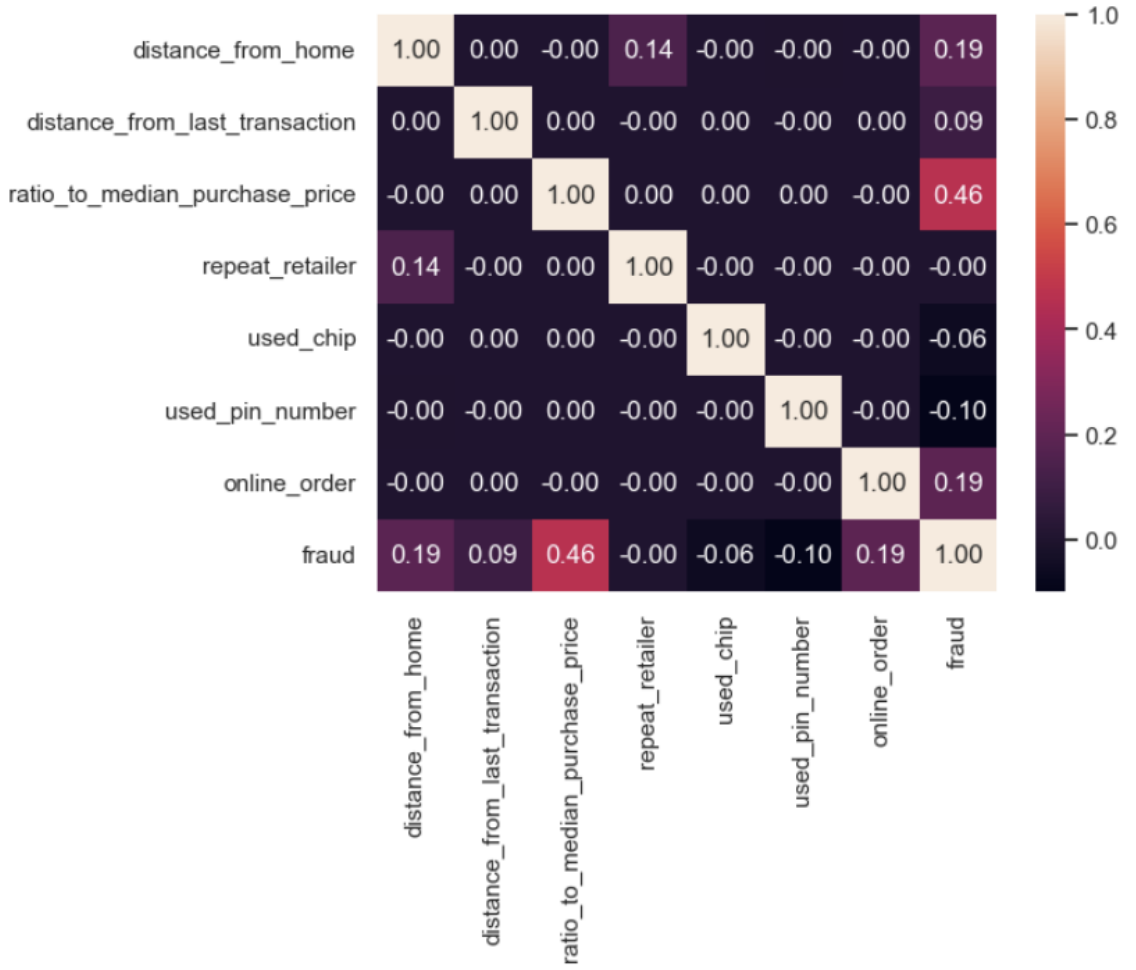

Fig. 1. Correlation matrix for dataset

4 METHODS

The overarching goal of this project is to be able to detect fraudulent transactions and protect consumers from those fraudulent transactions. To achieve this goal, two binary classifiers were trained, a deep neural network model and a random forest model, and the outcomes of these models were compared. For each of these models, classification accuracy was the most important metric while also minimizing undetected fraudulent transactions as much as possible.

The dataset utilized in this project contained data that was already cleaned and nearly ready to be used out of the box. The dataset contained no duplicates nor any rows with any missing information. The StandardScaler from the Scikit-Learn package was utilized to perform normalization on the data before training to reduce impacts of outliers on the model and increase classification accuracy. Table 3 shows one row of training data before and after the StandardScaler transformation.

Feature Name	Before Normalization	After Normalization
distance_from_home	5.38	-0.34
distance_from_last_transaction	2.05	-0.15
ratio_to_median_purchase_price	0.84	-0.46
repeat_retailer	1	0.37
used_chip	0	-0.71
used_pin_number	0	-0.30
online_order	1	0.65

Table 3. One row of training data before and after StandardScaler normalization (rounded to 2 decimal places)

As mentioned previously, one of the challenges with training machine learning algorithms to detect credit card fraud is the overwhelming majority of transactions that occur are legitimate, and only 9% of rows of the selected dataset contain fraudulent transactions. In an effort to balance the training and prevent biasing, an initial subset of the data was selected to contain all of the fraudulent transactions along with an equal number of randomly selected legitimate transactions. This created a dataset containing 50% fraudulent transactions and 50% legitimate transactions which was anticipated to achieve the best results in testing to prevent biasing. While this was successful to a degree and achieved some meaningful results, both models struggled with too many misclassifications of legitimate transactions as fraudulent transactions. Through some trial and error, it was determined that a split of 25% fraudulent transactions and 75% legitimate transactions yielded the best results for training both models, and this is what was utilized for both final models. Once the optimal dataset had been discovered, we were able to determine the optimal test, validation, and training data sizes. Ultimately, we split the dataset into 40% test data, 15% validation data, and 45% training data for the Deep Neural Network model and 30% test data, 17.5% validation data, and 52.5% training data for the Random Forest Model.

4.1 Deep Neural Network Model

The Deep Neural Network model was created utilizing the Keras library in Python. The network was configured with a total of 4 layers which were all densely connected. The first 3 layers contained 64 nodes each and all utilized the Rectified Linear Unit (ReLU) activation function. The fourth layer is a single node densely connected with a sigmoid activation function which is necessary to perform binary classification. In between each layer of nodes a 20% dropout is included to help prevent the model from over-fitting. This network configuration was determined to provide the best results for this implementation based on experimental results and iterative experimentation. The model was compiled and trained utilizing the Adam optimizer at a learning rate of 0.00025, utilizing binary cross entropy for the loss function, and focusing on accuracy as the primary metric for 20 epochs. The learning rate and epoch count were also found to provide the best results over the experimentation process.

4.2 Random Forest Model

The Random Forest model is provided directly by the scikit-learn library in Python. Previous research has been performed with random forest models and has shown to typically work well with our type of dataset. The model contains an algorithm that takes in each of the inputs and determines where and when to split into a new decision tree. Our model was configured to not be limited to the max number of leafs in the overall tree. The accuracy had shown to be the best with allowing the default model to determine the number of leafs needed with the given input data. The

Random Forest model used the same training and test data as the Deep Neural Network model, including normalizing the data. However, decision trees are not impacted by non-normalized data, but there was a need to normalize due to internal requirements for the Random Forest model in Python.

5 RESULTS

5.1 Deep Neural Network

Fig. 2. Neural Network training and validation accuracy progression over epochs

Fig. 3. Neural Network training and validation loss progression over epochs

Both Figures 2 and 3 show the model history as it trains over a period of 20 epochs for accuracy and loss respectively. Each graph contains both a plot of the training metric and the validation metric. The key difference between these two metrics is that the machine learning model learns primarily from the training set, which gives the training accuracy or

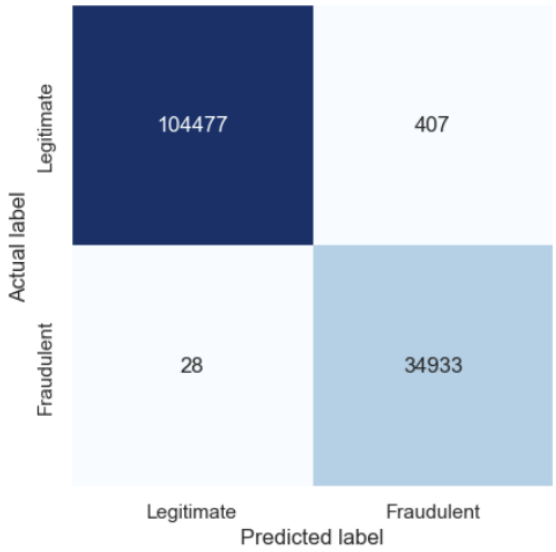

Fig. 4. Neural network's binary classification confusion matrix for test set from split dataset

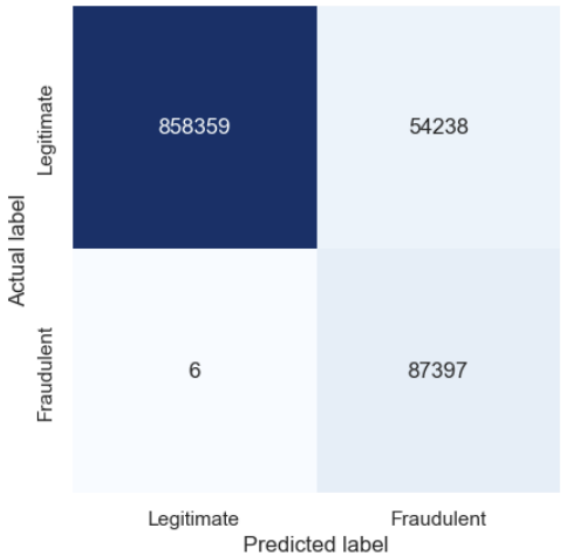

Fig. 5. Neural network's binary classification confusion matrix for running model on full original dataset

loss, while the machine learning model looks to validate its approach based on the feedback it receives from validation set. The model achieves over 99.5% training and validation accuracy at the twentieth epoch.

Figure 4 shows the deep neural network model's performance against the test portion of the split dataset in the version of a confusion matrix. Figure 5 shows the deep neural network model's performance against the entire original dataset. When the model was presented with the full dataset, it achieved an overall accuracy of approximately 94.5% in

correctly labeling transactions as either fraudulent or legitimate. The model only mislabeled 6 fraudulent transaction as legitimate, but it also mislabeled 54,238 legitimate transactions as being fraudulent.

5.2 Random Forest

The Random Forest model easily achieved an accuracy of 95% and we were able to train the model to minimize the number of fraudulent transactions being misclassified. There were generally between 2 and 8 fraudulent transactions that were misclassified as legitimate. However, there were 48,754 or more transactions that were misclassified as fraudulent. This could possibly be reduced if there was a larger number of actual fraudulent transactions in the dataset. The model would have more fraudulent transactions to make decisions on whether more or less leaf nodes were to be needed during training. Optimizing the decision trees overall would result in better decisions from each of the trees in the model. Therefore, resulting in better overall decisions for whether a given transaction is fraudulent or not. Where the number of transactions that were misclassified as fraudulent is pretty high, we were still able to keep our goal of minimal fraudulent transactions being misclassified.

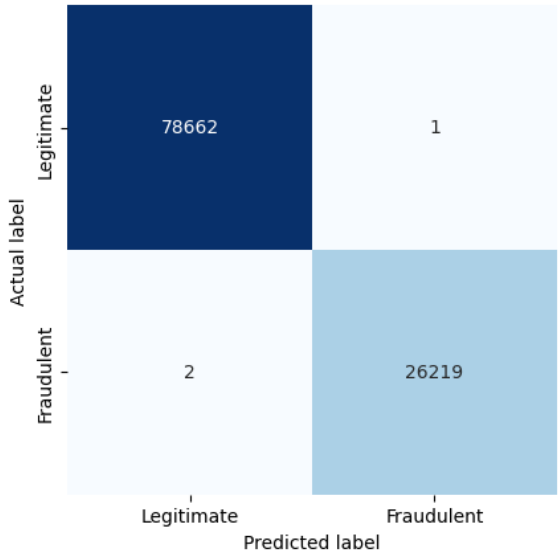

Fig. 6. Random forest's binary classification confusion matrix for test set from split dataset

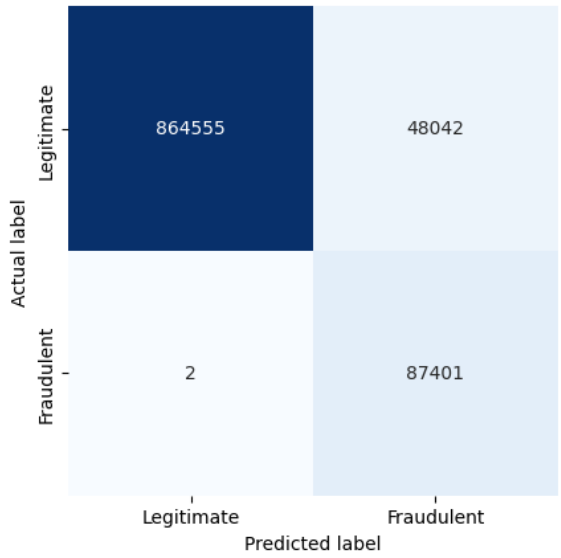

Fig. 7. Random forest's binary classification confusion matrix for running model on full original dataset

6 DISCUSSION

The project sought to determine whether a machine learning model could be successfully trained to detect fraudulent credit card transactions. Success being defined as the capability of correctly identifying transactions as fraudulent or non-fraudulent approximately 95% of the time. We found that not only can a machine learning model be trained to detect fraudulent transactions,but it was more likely to label a non-fraudulent transaction as fraudulent than it was to misclassify a truly fraudulent transaction. As this is a model that would be intended for financial institutions, detecting as many fraudulent transactions as possible, even when having to manually investigate mislabeled non-fraudulent transactions, is likely to be seen as the optimal outcome.

While the training and prediction ability of the models was successful, it still remains apparent that the dataset was sub-optimal for the task. It follows that a larger dataset, with the fraudulent transactions accounting for more than 10% of the data, would allow the model more opportunity to learn the patterns of the two categories. Hence, being able to differentiate between them more accurately and lower the total number of misclassified non-fraudulent transactions. Finding such a dataset in the time we were given to complete the project ultimately proved impractical. Furthermore, as the dataset was obtained from a dataset collections website, verifying the authenticity is nearly impossible. So, while it was a dataset that was easy to work with, it is unsuitable for use in a more paramount application.

Our research and development efforts have provided valuable insights, yet, there are still more avenues for future exploration. Given less of a time constraint we would explore other viable models for credit fraud detection, such as Support Vector Machines and Auto-encoders. In future endeavors, we would like to experiment with training using partially-synthetic data, utilizing a tool such as SMOTE, as a way to increase the number of fraudulent samples in a training set as well as optimizing our parameters throughout the models.

7 CONCLUSION

We have found that machine learning is indeed a viable solution for detecting credit fraud. Both the Random Forest and Deep Neural Network models were able to achieve a 95% accuracy when tested against the entire dataset. The Random Forest slightly outperformed the Deep Neural Network model, due to the lower number of misclassified legitimate transactions. However, either model can detect a fraudulent transaction in over 99% of cases.

REFERENCES

[1] Yashvi Jain, Namrata Tiwari, Shripriya Dubey, and Sarika Jain. 2019. A comparative analysis of various credit card fraud detection techniques. *International Journal of Recent Technology and Engineering* 7, 5 (2019), 402–407.

[2] Navanshu Khare and Saad Yunus Sait. 2018. Credit card fraud detection using machine learning models and collating machine learning models. *International Journal of Pure and Applied Mathematics* 118, 20 (2018), 825–838.

[3] John S. Kiernan. 2024. *Credit Card Fraud Statistics.* WalletHub. Retrieved May 9, 2024 from https://wallethub.com/edu/cc/credit-card-fraud-statistics/25725

[4] Capital One. 2023. *Number of Credit Card Transactions per Second, Day, & Year.* Capital One. Retrieved May 9, 2024 from https://capitaloneshopping.com/research/number-of-credit-card-transactions/

[5] Dhanush Narayanan R. 2022. Credit Card Fraud, Version 1. Retrieved 4-11-2024 from https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud/data

[6] V Sahayasakila, D Aishwaryasikhakolli, V Yasaswi, et al. 2019. Credit card fraud detection system using smote technique and whale optimization algorithm. *International Journal of Engineering and Advanced Technology (IJEAT)* 8, 5 (2019), 190.

[7] Ruttala Sailusha, V. Gnaneswar, R. Ramesh, and G. Ramakoteswara Rao. 2020. Credit Card Fraud Detection Using Machine Learning. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS).* 1264–1270. https://doi.org/10.1109/ICICCS48265.2020.9121114

[8] Unisys. 2021. 2021 Unisys Security Index. Retrieved 5-9-2024 from https://www.unisys.com/siteassets/microsites/unisys-security-index-2021/report-usi-2021.pdf

Received 10 May 2024