Environment. Technology. Resources. Rezekne, Latvia
Proceedings of the 16th International Scientific and Practical Conference. 2025. Volume V, 255-263

Exploring the Potential of Artificial Intelligence to Predict Cyber Attacks: Creation, Evaluation and Comparative Analysis of Effective Models of Fine-Tuning, Random Forests, and Neural Networks

Miroslav Stefanov
Computer science department
ULSIT
Sofia, Bulgaria
m.stefanov@unibit.bg

Boyan Jekov
Computer science department
ULSIT
Sofia, Bulgaria
b.jekov@unibit.bg

Tito Titov
Computer science department
ULSIT
Sofia, Bulgaria
t.titov@unibit.bg

Andrian Stoilov
Computer science department
ULSIT
Sofia, Bulgaria
a.stoilov@unibit.bg

Kiril Nikolov
Computer science department
ULSIT
Sofia, Bulgaria
k.nikolov@unibit.bg

***Abstract*— This quantitative investigation focuses on the application of artificial intelligence (AI) models for predicting cyberattacks and detecting anomalies in network traffic, aiming to enhance cybersecurity defenses. With the increasing complexity of cyber threats, AI offers a promising solution to address these challenges by providing predictive and responsive capabilities. This study compares three AI models — Fine-Tuning, Random Forests, and TensorFlow — using datasets aggregated on daily, weekly, and monthly levels. The methodology includes advanced data preprocessing, statistical analysis, and evaluation metrics such as RMSE, R², Precision, Recall, and F1-Score. Random Forests demonstrated exceptional accuracy and reliability, achieving high R² values and minimal errors. Fine-Tuning showed strong predictive capabilities but required careful parameter tuning to maintain accuracy. TensorFlow proved to be a powerful tool but required optimization to improve precision and reduce false positives. These results highlight the importance of model selection and parameter tuning in AI-driven cybersecurity applications.**

Keywords*— *Anomaly Detection, Artificial Intelligence, Cyber Attack Prediction, Cybersecurity, Machine Learning Models

I. INTRODUCTION

In the digital age, cybersecurity has become critical due to the exponential increase in cyberattacks, threatening individuals, organizations, and global infrastructure [1]. The increasing complexity and frequency of threats often exceed the capabilities of conventional defense mechanisms. Artificial Intelligence (AI), with its predictive and analytical strengths, offers promising solutions to these challenges [2]. This study investigates the potential of AI methodologies, specifically focusing on Fine-Tuning, Random Forests, and TensorFlow, to predict and mitigate cyber threats through anomaly detection and threat prediction.

The integration of Artificial Intelligence (AI) into cybersecurity highlights the need for adaptive and intelligent solutions to manage the dynamic nature of cyber threats [2]. Among AI-driven models, Random Forests and TensorFlow have shown particular promise. For example, Quezada et al [3] employed Random Forests in a bot infection detection system leveraging DNS fingerprinting, demonstrating the model's efficiency and adaptability in real-time anomaly detection. TensorFlow and deep learning models, as reviewed by Do et al. [4], have proven

Online ISSN 2256-070X
https://doi.org/10.17770/etr2025vol5.8504

© 2025 The Author(s). Published by RTU PRESS.
This is an open access article under the Creative Commons Attribution 4.0 International License.

effective in phishing detection through innovative approaches like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. However, these methods face challenges such as the need for extensive training data and optimization to improve accuracy and scalability.

Despite these advancements, gaps remain in addressing issues like data availability and computational costs. This study evaluates the strengths and limitations of Fine-Tuning, Random Forests, and TensorFlow, focusing on their adaptability to evolving cybersecurity threats.

AI models have demonstrated substantial success in cybersecurity applications, particularly in predictive analytics and anomaly detection. Lichy et al. [5] highlight that classical machine learning models like Random Forest often outperform deep learning models such as Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) in contexts like classifying encrypted malware traffic. Random Forest's robustness in structured data and lower computational complexity make it highly effective for real-time cybersecurity applications. However, TensorFlow and deep learning models excel in unstructured data contexts but face challenges such as susceptibility to overfitting and high computational demands. These limitations underscore the importance of selecting appropriate models based on specific scenarios.

This study evaluates the strengths of Fine-Tuning, Random Forests, and TensorFlow, focusing on their adaptability to cybersecurity applications and their ability to address operational constraints effectively.

Extensive research on AI applications in cybersecurity has highlighted methods for anomaly detection, DDoS attack prevention, and phishing detection. However, there is a notable gap in comparative studies evaluating diverse AI models on aggregated and temporally segmented datasets [6]. This study addresses this gap by assessing the performance of Fine-Tuning, Random Forests, and TensorFlow.

Fine-Tuning adapts pre-trained models to specific tasks by retraining with new data, leveraging existing knowledge to reduce computational and data requirements [7]. Random Forests, an ensemble method, improves classification and regression accuracy by reducing tree correlation through "bagging" and random feature selection [8]. TensorFlow, an open-source machine learning library, enables the creation and training of neural networks for complex computational tasks [9]. These models are evaluated for their adaptability to varied temporal datasets and operational constraints.

II. Materials and methods

The analytical process in this qualitative research utilized widely recognized tools in scientific research, including Jupyter Notebook and Python. Jupyter Notebook was employed for its interactivity and ability to facilitate detailed data analysis. For analyzing models like Fine-Tuning, Random Forests, and TensorFlow, Python libraries such as Pandas were used for efficient data manipulation, while Matplotlib and Seaborn enabled clear and intuitive graphical visualization [10]. Scikit-learn provided advanced statistical models and machine learning algorithms, which were crucial for evaluating these models [11].

This team ensured the reliability of the analysis by conducting metadata standardization and validation procedures to identify anomalies that could influence results. Statistical methods such as mean, median, mode, standard deviation, and extreme value calculations were applied to provide a comprehensive understanding of the data distribution, which was critical for assessing these models in cybersecurity contexts.

The research methodology outlines the structured approach used to investigate the effectiveness of various AI models in cybersecurity applications. By leveraging robust platforms like Python and Jupyter Notebook, the research integrates advanced libraries for statistical computations, visualization, and machine learning. The methodology emphasizes data preprocessing, consistent training and testing practices, and rigorous performance assessment to ensure reliable results.

The analytical process in this research is organized using tools and platforms widely applicable in scientific studies, including Jupyter Notebook and Python. Jupyter Notebook offers an interactive environment for detailed data analysis, enhancing the workflow for evaluating models such as Fine-Tuning, Random Forests, and TensorFlow. Python libraries like Pandas are utilized for efficient data manipulation, while Matplotlib and Seaborn enable clear and intuitive graphical visualizations [10]. Scikit-learn is employed to apply advanced statistical models and machine learning algorithms [11], ensuring a robust framework for analyzing the performance of these models in cybersecurity.

This research team carried out metadata standardization and validation procedures to ensure the reliability of the analytical process. These steps were crucial in identifying and addressing any anomalies or exceptions that could impact the evaluation of models such as Fine-Tuning, Random Forests, and TensorFlow. The statistical analysis involved calculations for central tendency and dispersion, including means, medians, modes, standard deviations, and extreme values. These metrics provided a comprehensive understanding of the data distribution, supporting a robust assessment of the models' performance in cybersecurity contexts.

Analyzing cyberattacks based on aggregated data from various time periods provides critical insights into trends and patterns, essential for enhancing cybersecurity strategies. This research utilized three main datasets—Monthly, Weekly, and Daily—each contributing valuable information for evaluating the effectiveness of Fine-Tuning, Random Forests, and TensorFlow. The datasets were derived from actual cyberattacks, collected through the implementation of a honeynet system across multiple municipalities in Bulgaria.

To ensure compliance with legislative requirements, all data were anonymized and used exclusively for research purposes. The honeynet system simulated cyberattacks, capturing detailed information on attack behaviors and trends. This foundational data enabled the analysis of these AI models, supporting the development of predictive tools and improving defense strategies in cybersecurity.

Cyberattack data was analyzed using daily, weekly, and monthly temporal breakdowns to identify patterns and trends at different levels of granularity. This segmentation supported the development of AI models like Fine-Tuning, Random Forests, and TensorFlow, enabling them to perform short-term anomaly detection, medium-term trend analysis, and long-term attack predictions. By aligning temporal resolutions with specific analytical objectives, the study provided a robust framework for understanding and mitigating cybersecurity threats.

The dataset for daily analysis spanned 200 days, during which 196,494,832 attacks were recorded from all IP addresses, including 25,251,071 originating from Bulgarian IPs. On average, each honeypot registered 17,863,166 attacks, with attacks from Bulgarian IPs comprising 12.86% of the total. This data enabled models like TensorFlow and Random Forests to identify patterns and trends in daily attacks, enhancing their ability to detect anomalies and predict future cyber threats effectively.

A. File: Weekly

The Weekly dataset spans 20 months, equivalent to approximately 87 weeks, and includes a total of 196,494,832 recorded attacks from all IP addresses. Of these, 25,251,071 attacks originated from Bulgarian IPs, accounting for 12.86% of the total. On average, 113,006 attacks from all IPs and 14,046 attacks from Bulgarian IPs were registered daily, with foreign attacks surpassing Bulgarian ones by a factor of 6.78. The models utilized periodic attack patterns to build prevention strategies and strengthen anomaly detection capabilities. By differentiating between attacks from foreign and Bulgarian IPs, these models developed more precise detection mechanisms for identifying internal and external threats effectively.

B. File: Months

The monthly dataset spans 10 months and includes a total of 352,376,021 recorded attacks from all IP addresses. Of these, 26,004,126 attacks originated from Bulgarian IPs, accounting for 7.38% of the total. On average, 35,237,602 attacks from all IPs and 2,600,413 attacks from Bulgarian IPs were logged daily, with 4,783 unique Bulgarian IP addresses identified during this period. By analyzing geographic characteristics and network topology, these models classified attack types and distinguished IP origins. This approach enabled the models to detect and predict large-scale patterns effectively, addressing both local and global cybersecurity threats.

All collected data were meticulously anonymized to adhere to legislative and ethical standards, ensuring the integrity of the research. The anonymization process involved removing or masking any personally identifiable information (PII) and sensitive metadata to eliminate privacy risks. These datasets were utilized exclusively for research purposes, enabling the evaluation of models, while safeguarding participants' privacy and fostering robust cybersecurity analysis.

C. Software Tools for Analysis

The analysis and evaluation of cyberattack data in this study required the use of advanced software tools to ensure accuracy and efficiency. Python served as the primary platform for implementing models like Fine-Tuning, Random Forests, and TensorFlow, supporting data manipulation, preprocessing, and machine learning tasks. Key libraries, including Pandas for data handling [12], Matplotlib and Seaborn for visualization [10], [13], and Scikit-learn for machine learning algorithms [11], provided robust analytical capabilities.

Jupyter Notebook offered an interactive environment that enhanced reproducibility and iterative experimentation, critical for refining these models' workflows [14]. These tools enabled the systematic analysis of cyberattack patterns and facilitated comprehensive evaluations of the models’ predictive performance, contributing to advancements in cybersecurity applications.

Pandas was integral to the research process, supporting high-performance data manipulation and preprocessing for models. It was used to load and structure large datasets, including network traffic logs and metadata on cyberattacks, ensuring the data was well-organized for further analysis. The tool supported cleaning processes, such as removing missing values and duplicate entries, and normalized numerical features to ensure compatibility with machine learning algorithms. Furthermore, Pandas facilitated the splitting of data into training and testing subsets, essential for evaluating these models' performance. Its exploratory data analysis capabilities helped identify key trends and anomalies in the initial dataset, providing critical insights for building effective machine learning workflows [12].

Matplotlib and Seaborn were essential tools for visualizing data and interpreting results in the evaluation. Matplotlib was used to create scatterplots and boxplots, enabling the visualization of data distributions and the identification of potential outliers. Seaborn complemented these capabilities by generating heatmaps and KDE plots, which highlighted correlations and local trends among network traffic features. These visualizations were instrumental in guiding the selection of appropriate anomaly detection algorithms and in understanding the key outputs of these models. By providing clear graphical representations, Matplotlib and Seaborn enhanced the analytical process and supported informed decision-making throughout the research [10].

Scikit-learn was a central tool for implementing and evaluating the performance of AI models like Fine-Tuning, Random Forests, and TensorFlow. It enabled the integration of advanced methods for anomaly detection and predictive analytics, ensuring seamless implementation and evaluation workflows. The library provided automated computation of key metrics such as Precision, Recall, F1-score, and RMSE, which were critical for assessing model performance. This comprehensive functionality made it indispensable for refining these models and enhancing their effectiveness in cybersecurity applications [11].

Jupyter Notebook was a central platform for executing, documenting, and visualizing the research process, particularly for thee models. Each stage of the analysis was organized into individual cells, enabling iterative testing and optimization of these models and their associated hypotheses. The interactive environment provided visual feedback through embedded graphs and tables, facilitating dynamic parameter tuning and improving the efficiency and clarity of the analysis [14].

D. Comparison of Methods

Three AI models Fine-Tuning, Random Forests and TensorFlow (neural networks), were selected for this study to evaluate their effectiveness in predicting cyberattacks and detecting anomalies in network traffic. The study was done in three steps. The first one was Training and Testing - Each model was trained and tested on the same datasets to ensure consistency in comparisons. The second one was Handling Model Specifics - Hyperparameter optimization was applied to neural networks, such as TensorFlow. And the last one was Performance Assessment - Each model's effectiveness was evaluated using key metrics (see the "Metrics section), which measure both predictive and classification performance.

Three primary datasets Days, Months, and Weekly, were utilized to provide varying temporal contexts for analyzing cyberattack patterns. Data preprocessing involved removing missing or duplicate values and normalizing or standardizing features where needed. For models like TensorFlow, the datasets were split into training, validation, and test subsets, ensuring robust evaluation and optimization.

The analysis goals varied for each dataset: daily data enabled time-series analysis to identify short-term trends, while monthly data offered a long-term perspective on global anomalies. Weekly data supported the discovery of periodic patterns and comparisons between internal (Bulgarian) and external (foreign) threats. This structured approach allowed TensorFlow, Random Forests, and Fine-Tuning to adapt to diverse cybersecurity challenges effectively.

The effectiveness of the models was evaluated using a range of metrics tailored to their functionalities. RMSE measured the average error, making it ideal for regression models like TensorFlow, while R^2 assessed the proportion of data variability explained by each model, indicating their predictive accuracy. Precision and Recall were used to evaluate classification tasks, focusing on the models' ability to identify cyberattacks and detect true positives accurately. F1-Score provided a harmonic mean of Precision and Recall for a balanced evaluation.

To ensure the robustness of the results, cross-validation was employed for all models, with repeated tests conducted on varied subsets. This approach minimized the risk of overfitting and improved the generalizability of Fine-Tuning, Random Forests, and TensorFlow. Consistent validation practices ensured reliable performance assessments for these models.

After applying each model to the datasets, the results were compared using the outlined metrics. Fine-Tuning demonstrated high accuracy but was sensitive to minor changes in the data. Random Forests showed perfect metrics, though results may indicate potential overfitting. TensorFlow showed that significant hyperparameter optimization is required for better results.

III. Results and Discussion

The following chapter focuses on the analysis and evaluation of AI models, including Fine-Tuning, Random Forests, and TensorFlow, applied to cybersecurity. The evaluation uses performance metrics five to measure the models' effectiveness in predicting cyberattacks and detecting anomalies. These analyses provide valuable insights into selecting and optimizing AI models based on specific cybersecurity objectives and data characteristics. The findings emphasize the importance of tailoring AI methodologies to align with real-world cybersecurity challenges, enhancing the models' effectiveness in anomaly detection and predictive defense strategies.

The Fine-Tuning Model demonstrates consistent predictive capabilities, as evidenced by the RMSE (Root Mean Square Error) value of 726,738.906. While this value indicates some deviations from actual data, it remains within reasonable limits for models applied to large and complex datasets. Performance metrics suggest that the model is suitable for applications requiring high accuracy in predictive tasks and analytical processing. The results presented in Table 1were achieved through a systematic testing and validation process that ensures the model's reliability. The methodology is detailed below:

The data was divided into training (80%) and testing (20%) subsets. The training set was used to build the model, while the testing set provided an independent assessment of the model's generalization capability.

During training, the k-fold cross-validation method was applied. This approach ensured consistent model performance across different data partitions and reduced the likelihood of overfitting. It also facilitated the optimization of hyperparameters.

Prior to training, the data was balanced to ensure equal representation of all classes. This step minimized biases in classification metrics, particularly precision and recall.

The model underwent hyperparameter tuning using methods such as grid search. This process aimed to minimize RMSE and improve classification metrics, ensuring the model's reliability across different configurations.

Test RMSE: Calculated to quantify the average prediction error. This metric penalizes larger deviations more heavily, making it an effective indicator of predictive accuracy.

Test R²: The coefficient of determination, with a value of 0.985, highlights the model's ability to explain data variation. This high R² value indicates that the model fits the data well.

Precision, Recall, and F1 Score: These metrics were calculated to evaluate the model's classification performance. Scores of 1.0 for all these metrics suggest that the model effectively distinguishes between classes without false positives or negatives.

Residual errors (differences between predicted and actual values) were analyzed to confirm their random distribution. This step validated the model’s predictions and ensured the absence of systematic biases.

Results were validated on the independent test dataset (20%), which was not used during training or hyperparameter tuning. This ensured an unbiased evaluation of the model’s performance.

TABLE 1 TABLE OF TEST RESULTS OF FINE-TUNING MODEL

Metric	*Value*
Test RMSE	726,738,906
Test R²	0.985
Precision	1.0
Recall	1.0
F1 Score	1.0

By adhering to this testing and evaluation process, the model demonstrates its robustness. Future work could involve applying the model to more diverse datasets to confirm its generalization capabilities and enhance its predictive accuracy.

Although the RMSE value is relatively high, possibly due to the complexity of the data or the specifics of the study, the Fine-Tuning Model exhibits reliable performance in predictive and classification tasks. It is well-suited for applications requiring precision and detailed analytical capability. As such, this model becomes a valuable tool in the field of data analysis, particularly in the context of malware detection and network traffic anomaly detection, where accuracy and prediction reliability are key.

The results in Table 2 for the Random Forests model were achieved through a systematic testing process, which aimed to evaluate the model's performance comprehensively. See Table 2 for details. Below are the detailed steps of the methodology and considerations for interpreting the outcomes:

The dataset was divided into training and testing sets in an 80-20 split. The training set was used to build the model, while the test set was reserved for an unbiased evaluation of the model's performance.

The training set underwent k-fold cross-validation to enhance reliability. This step helps ensure that the model performs consistently across different subsets of the data and minimizes the risk of overfitting.

Grid search and random search techniques were applied to optimize key hyperparameters, such as the number of trees, maximum depth, and splitting criteria. These optimizations were aimed at maximizing predictive accuracy while preventing overfitting.

The dataset was checked and adjusted for class balance to prevent bias toward the majority class. Techniques such as oversampling the minority class or undersampling the majority class were applied where necessary.

Test RMSE: The Root Mean Square Error measures the average error magnitude between the predicted and actual values. An RMSE of 0.0 may suggest potential data or evaluation anomalies.

R² Score: A score of 1.0 reflects that the model explains 100% of the variance in the target variable, which aligns with the RMSE of 0.0.

Precision, Recall, and F1 Score: All these metrics recorded perfect values of 1.0, signifying flawless classification performance.

TABLE 2 TABLE OF TEST RESULTS OF RANDOM FORESTS

Metric	*Value*
Test RMSE	0.0
Test R²	1.0
Precision	1.0
Recall	1.0
F1 Score	1.0

Residual Analysis:

The results presented in the table demonstrate the performance of the Random Forests model, with Precision, Recall, and F1 Score reaching values of 1.0. Additionally, RMSE is reported as 0.0, and R² has a value of 1.0. However, these results require further analysis to ensure their reliability and identify potential issues such as overfitting or data anomalies.

The model must be tested on an independent dataset that was not used during training or testing to validate the results. Conducting tests with more complex and diverse datasets is necessary to assess the model’s ability to generalize to real-world data. As part of the analysis, it should be noted that the results reported in the table may indicate overfitting. This suggests that the model may be overly tailored to the training data and fail to generalize effectively to new, unseen data, leading to unrealistic performance metrics.

The possibility of data leakage, where test data may have influenced the training process, must be considered. Additionally, an overly simplistic dataset could lead to exaggerated results, underscoring the need for using complex and realistic datasets. If the model bases its predictions on irrelevant or non-essential features, it could compromise its ability to perform effectively on new and unseen data.

Analyzing the importance of features is crucial to ensure that the model relies on relevant data rather than noise. The perfect results could also stem from statistical errors or anomalies in the testing process, which requires a thorough review of residual errors and key metrics.

Random Forests perform well in the analysis of large datasets with many features, demonstrating their capabilities in both classification and regression analysis. They can be used to classify days with high and low probability of attacks based on historical data.

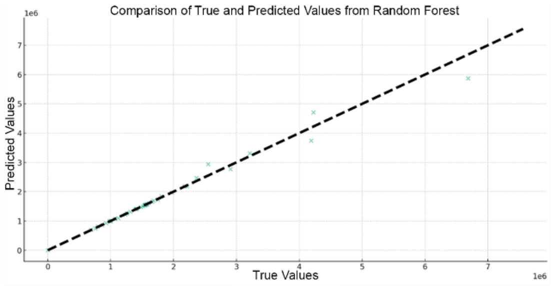

Fig. 1. Graph: Comparison of true and predicted values from Random Forest

The Figure 1 graph shows a comparison between the true values (X-axis) and the predicted values (Y-axis) for daily cyber attacks. The dashed line represents the ideal correlation, where the predicted values perfectly match the true values. Most points are aligned close to this ideal line, indicating that the model successfully captures the overall trends in the dataset. However, the following two points show noticeable deviations, highlighting areas where the model's predictions differ from the actual data.

Mean Squared Error (MSE): MSE is approximately 31,992,930,846. This high value reflects the quadratic scale of deviations and is influenced by the large magnitude of the target variable (daily cyber attacks) and the inherent variability in the dataset. It is important to note that MSE is sensitive to outliers, which can further increase this value.

Root Mean Squared Error (RMSE): RMSE is 178.866, providing a more intuitive measure as it is in the same unit as the target variable. This value suggests a reasonable average deviation between the predicted and true values, considering the high variability of the cyber attack data.

The average number of attacks recorded per day is averaged based on the number of honeypots deployed in the investigated areas. This averaging of attacks per honeypot per day is crucial for providing an accurate and representative analysis. The reason for using this approach is to minimize the impact of differences in the number of active honeypots during the observation period and to ensure uniformity in the presented data.

Based on multiple decision trees, random forests incorporate randomness in the training of submodels to reduce the variance of predictions and improve overall predictive power.

In the context of cyber security, random forest models allow analysis of the interaction between different variables and their influence on the probability and intensity of cyber attacks. Thanks to their ability to combine hundreds or even thousands of individual decision trees, these models can provide deep insights into data characteristics and predict potential attacks with high accuracy.

When using random forests for regression tasks, the model shows robust results as measured by RMSE, which allows an accurate estimation of the mean prediction error. For classification tasks, the model can be evaluated with an F-score, which provides a balanced measure of the model's accuracy and responsiveness to different classes. However, the F-score must be calculated in the context of clearly defined classification criteria that meet the specific needs of the cybersecurity problem.

Random forests are proving to be a reliable and flexible tool for cybersecurity analysis and prediction. Their ability to process large data sets and identify complex patterns makes this approach valuable for developing effective defense strategies and preventing future attacks. Future research can focus on fine-tuning the model and exploring integration opportunities with other ensemble and machine-learning techniques to further increase predictive accuracy and address the dynamically changing challenges in cyberspace.

Deep Learning can detect complex non-linear relationships in data and predict future attacks, making it a powerful tool for modeling intricate relationships. Recurrent Neural Networks (RNNs) are particularly well-suited for time series and sequential data, allowing for various applications such as prediction, classification, regression, and even data generation. Neural networks are versatile and can be utilized for a wide range of tasks, including classification, regression, time series forecasting, and more. When the data exhibits complex non-linear relationships, deep learning approaches, particularly RNNs, can be effective for analyzing attack data. See Table 3 for metrics and values.

TensorFlow is an open source machine learning software library that provides tools for building and training neural networks to recognize and discover patterns and correlations in large data sets. This open source machine learning software library is suitable for a wide range of tasks including regression, classification, automatic text generation, image processing and many others. Neural networks are particularly powerful in extracting complex features from data and are suitable for irregular data where relationships can be non-linear and complex. See Table 3 for details.

TABLE 3 TABLE OF TEST RESULTS OF TENSORFLOW

Metric	***Value***
Test RMSE	1,890,971,414
Test R²	0.9018
Precision	0.0
Recall	0.0
F1 Score	0.0

On the graph, we see the distribution of predicted values versus true values. The dashed red line represents the perfect fit where the predicted values equal the true values. Dots represent actual model predictions; the closer they are to the red line, the more accurate the predictions.

Based on the RMSE and R² values you provided:

RMSE (Root Mean Squared Error): 1,890,971.414 - This is a relatively high value, which means that your predictions are significantly off the true values. Ideally, we want the RMSE to be as low as possible.

R² (Coefficient of Determination): 0.9018 - This is a fairly high value indicating that the model explains about 90% of the variation in the true values. However, the high value of RMSE indicates that despite the good fit of the data, the error variances are significant.

The graph shows that most of the predictions are close to the line of perfect fit, but there are some significant deviations.

This could be due to several factors:

The model may encounter challenges with certain 'outliers' or data points that significantly differ from the rest of the dataset. There could also be anomalies or extremely high or low values in the data that the model failed to predict accurately. Additionally, the dataset might lack sufficient diversity or contain hidden factors that the model was unable to capture.

In the context of regression, the Precision, Recall, and F-score metrics are not directly applicable, as they are defined for classification tasks. In regression, predictor values are continuous rather than discrete classes. However, if you want to convert the regression problem into a classification problem, you need to define a threshold where values above this threshold are considered one class (e.g. "attack") and values below it another (e.g. "not attack").

The TensorFlow model included in this analysis reveals significant challenges in its ability to make accurate and reliable predictions. With the highest RMSE (Root Mean Square Error) value of 1,890,971.414, this model has difficulties in accurately predicting the results, which is manifested in significant discrepancies between the predicted and actual values. A high RMSE value is an indicator of large prediction errors, which calls into question the reliability of the model in practical applications.

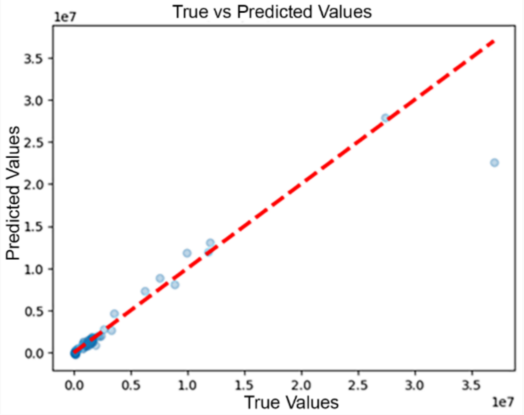

Fig. 2. Graph: Neural Networks - True versus predicted value

Furthermore, the R² value of 0.901, although relatively high, indicates that there is still a significant portion of the variation in the dependent variable that is not explained by the model. This suggests that the model may not include all relevant factors or interactions that are necessary for accurate prediction.

Most concerning are the zero Precision, Recall, and F-score values, which point to serious gaps in the model's ability to identify and classify relevant cases. This indicates that the model failed to correctly predict the positive cases, which is critical for tasks such as malware and network anomaly detection.

In conclusion, based on these results, the TensorFlow model requires significant improvements, including potentially retraining with more appropriate data or parameter tuning. It may also be necessary to investigate alternative models or techniques to achieve higher accuracy and reliability in predictions. At this stage, the model seems inadequate for the tasks at hand, especially in the context of cybersecurity and anomaly detection.

IV. CONCLUSIONS

In the context of regression, the Precision, Recall, and F-score metrics are not directly applicable, as they are defined for classification tasks. In regression, predictor values are continuous rather than discrete classes. However, if you want to convert the regression problem into a classification problem, you need to define a threshold where values above this threshold are considered one class (e.g. "attack") and values below it another (e.g. "not attack").

Table 4 contains the results of the different AI models:

TABLE 4 RESULTS OF THE AI MODELS

AI Model	***RMSE***	***R²***	***Precision***	***Recall***	***F-score***
Fine Tuning Model	726,738,906	0.985	1.0	1.0	1.0
Random Forests	0.0	1.0	1.0	1.0	1.0
TensorFlow	1,890,971,414	0.901	0	0	0

*The ranking presents the models from highest to lowest performance on these metrics.

This table presents a comparative analysis of different AI models, including statistics such as root mean square error (RMSE), coefficient of determination (R²), precision (Precision), range (Recall) and F-score, which allows evaluation of the effectiveness of each model.

The analysis of the results presented in the preceding table reveals significant differences in the performance of the different AI models in the context of the prediction or classification task within the studied data.

Fine Tuning Model: The Fine Tuning Model shows exceptional precision and stability as noted by the low RMSE value (726,738.906) and high coefficient of determination (R² = 0.985). This suggests that the model has an excellent ability to adapt to the data and generate accurate predictions.

Random Forests: The Random Forests model demonstrated a perfect score in all dimensions - RMSE (0.0), R² (1.0), Precision (1.0), Recall (1.0) and F-score (1.0). Such results may be indicative of an overtrained model, since a perfect result is rare in real-world applications.

TensorFlow: The TensorFlow model shows significant difficulty with an RMSE of 1,890,971.414 and an R² of 0.901, suggesting that the model may not be adequately trained or missing key features from the data.

In summary, the results emphasize the criticality of selecting a model that aligns with the characteristics of the data and the specific task requirements. While certain models demonstrate high precision and reliability, others may necessitate further tuning or refinement to enhance their predictive capabilities. Among the analyzed AI models, Random Forests stand out as the most effective, delivering exceptional accuracy and reliability across all evaluated metrics. With maximum Precision, Recall, and F-score values, alongside minimal RMSE and maximum R², Random Forests excel in both prediction and classification tasks, making them an optimal choice for applications demanding high accuracy. Conversely, the TensorFlow model faces notable limitations, rendering it less suitable for critical use cases that require stringent accuracy and reliability standards. Despite its widespread use, TensorFlow requires significant optimization and adjustments to match the performance levels of leading models like Random Forests.

Random Forests emerge as the optimal choice for complex analysis and prediction, demonstrating exceptional accuracy and reliability across diverse applications. This model consistently delivers robust and precise solutions for a broad range of tasks within machine learning and artificial intelligence. The analysis of the AI models highlights their effectiveness based on key criteria, including predictive accuracy (RMSE), the ability to explain data variability (R²), and classification performance (Precision, Recall, and F-score). These findings reinforce Random Forests as a highly dependable tool for addressing complex computational challenges.

This research examined various machine learning models, emphasizing their predictive and classification capabilities. The Random Forests model demonstrated exceptional performance, achieving perfect metrics, including a zero RMSE and a maximum R² value of 1.0, underscoring its robustness and reliability. The Fine-Tuning model also performed well, with high R² values and near-perfect Precision, Recall, and F-score metrics, though its RMSE was higher compared to Random Forests. In contrast, the TensorFlow model exhibited notable limitations, with significant discrepancies in prediction accuracy and low or zero Precision, Recall, and F-score values, highlighting challenges in classification tasks. This analysis provides an in-depth exploration of the strengths and weaknesses of various machine learning approaches, reinforcing the importance of selecting models tailored to specific tasks and datasets.

REFERENCES

[1] "How AI will automate cybersecurity in the post-COVID world | VentureBeat." Accessed: Feb. 25, 2025. [Online]. Available: https://venturebeat.com/business/how-ai-will-automate-cybersecurity-in-the-post-covid-world/

[2] "Hands-On Artificial Intelligence for Cybersecurity | Data | Paperback." Accessed: Feb. 25, 2025. [Online]. Available: https://www.packtpub.com/en-us/product/hands-on-artificial-intelligence-for-cybersecurity-9781789804027

[3] V. Quezada, F. Astudillo-Salinas, L. Tello-Oquendo, and P. Bernal, "Real-time bot infection detection system using DNS fingerprinting and machine-learning," *Computer Networks*, vol. 228, p. 109725, Jun. 2023, doi: 10.1016/J.COMNET.2023.109725.

[4] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," *IEEE Access*, vol. 10, pp. 36429–36463, 2022, doi: 10.1109/ACCESS.2022.3151903.

[5] A. Lichy, O. Bader, R. Dubin, A. Dvir, and C. Hajaj, "When a RF beats a CNN and GRU, together—A comparison of deep learning and classical machine learning approaches for encrypted malware traffic classification," *Comput Secur*, vol. 124, p. 103000, Jan. 2023, doi: 10.1016/J.COSE.2022.103000.

[6] S. Aktar and A. Yasin Nur, "Towards DDoS attack detection using deep learning approach," *Comput Secur*, vol. 129, p. 103251, Jun. 2023, doi: 10.1016/J.COSE.2023.103251.

[7] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North*, pp. 4171–4186, 2019, doi: 10.18653/V1/N19-1423.

[8] L. Breiman, "Random forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324/METRICS.

[9] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Mar. 2016, Accessed: Feb. 25, 2025. [Online]. Available: https://arxiv.org/abs/1603.04467v2

[10] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput Sci Eng*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[11] F. Pedregosa FABIANPEDREGOSA *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011, Accessed: Feb. 25, 2025. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[12] W. McKinney, "Data Structures for Statistical Computing in Python," *scipy*, pp. 56–61, 2010, doi: 10.25080/MAJORA-92BF1922-00A.

[13] M. L. Waskom, "seaborn: statistical data visualization," *J Open Source Softw*, vol. 6, no. 60, p. 3021, Apr. 2021, doi: 10.21105/JOSS.03021.

[14] T. Kluyver *et al.*, "Jupyter Notebooks – a publishing format for reproducible computational workflows," *Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016*, pp. 87–90, 2016, doi: 10.3233/978-1-61499-649-1-87.