



**FAKE NEWS CLASSIFICATION USING MACHINE LEARNING
AND DEEP LEARNING APPROACHES**

Saja Abdulhaleem Mahmood AL-OBAIDI

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING DEPARTMENT**

**GAZI UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

NOVEMBER 2023

ETHICAL STATEMENT

I at this moment declare that in this thesis study, I prepared the thesis writing rules of Gazi University Graduate School of Natural and Applied Sciences;

- All data, information, and documents presented in this thesis have been obtained within the scope of academic rules and ethical conduct,
- All information, documents, assessments, and results have been presented by scientific ethical conduct and moral rules,
- All material used in this thesis that is not original to this work has been exhaustively cited and referenced,
- No change has been made in the data used,
- The work presented in this thesis is original,

or else, I admit all loss of rights to be incurred against me.

Saja Abdulhaleem Mahmood AL-OBAID

08/12/2023

FAKE NEWS CLASSIFICATION USING MACHINE LEARNING
AND DEEP LEARNING APPROACHES

(M.Sc. Thesis)

Saja Abdulhaleem Mahmood AL-OBAIDI

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

November 2023

ABSTRACT

In the contemporary digital landscape, online news portals have become the primary information source for a diverse audience. However, the rampant spread of misinformation poses considerable challenges in discerning genuine news from fabricated narratives. This thesis undertakes an extensive examination of diverse machine learning (ML) and deep learning (DL) techniques to enhance the detection of fake news across various settings. Our research encompasses seven conventional ML methods, including Random Forest and Decision Tree, alongside five advanced DL approaches like LSTM and CNN. Rigorous testing was conducted on four distinct datasets: ISOT, FakeNewsNet, Dataset1, and Dataset2. A detailed analysis of each model's performance reveals notable variations in accuracy levels. For instance, CNN achieved a remarkable accuracy of 98.59% on the ISOT dataset, while CNN-GRU excelled on the FakeNewsNet dataset with an impressive 99.97% accuracy. In contrast, the LSTM-AutoEncoder model exhibited lower accuracy rates, ranging between 24% and 62%. Our study underscores the pivotal influence of dataset characteristics on model selection and optimization. By offering a comprehensive evaluation of ML and DL effectiveness in fake news detection, this research significantly contributes to advancing more potent strategies for combating online misinformation.

Science Code : 902.1.014
Key Words: : Fake news detection, machine learning, deep learning
Page Number : 219
Supervisor : Asst. Prof.Dr. TUBA CAĞLIKANTAR

MAKİNE ÖĞRENİMİ VE DERİN ÖĞRENME YAKLAŞIMLARINI KULLANARAK SAHTE HABER SINIFLANDIRMASI

(Yüksek Lisans Tezi)

Saja Abdulhaleem Mahmood AL-OBAIDI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Kasım 2023

ÖZET

Modern dijital ortamda, çevrimiçi haber portalları farklı bir izleyici kitlesi için birincil bilgi kaynağı haline gelmiştir. Ancak yanlış bilgilerin hızla yayılması, gerçek haberleri sahte haberlerden ayırmada önemli zorluklara neden olmaktadır. Bu tez, sahte haberlerin tespitini geliştirmek için çeşitli makine öğrenimi (ML) ve derin öğrenme (DL) tekniklerinin kapsamlı bir incelemesini sunmaktadır. Araştırmamız, Rastgele Orman ve Karar Ağacı dahil olmak üzere yedi geleneksel makine öğrenimi yöntemini ve ayrıca LSTM ve CNN gibi beş gelişmiş DL yaklaşımını kapsamaktadır. ISOT, FakeNewsNet, Dataset1 ve Dataset2 olmak üzere dört farklı veri kümesi üzerinde detaylı testler gerçekleştirilmiştir. Her modelin performansının ayrıntılı analizi, doğruluk seviyelerindeki dikkate değer farklılıkları ortaya çıkarmıştır. Örneğin, CNN, ISOT veri kümesinde %98,59 gibi dikkate değer bir doğruluk elde ederken, CNN-GRU, FakeNewsNet veri kümesinde %99,97 gibi önemli bir doğruluk oranıyla öne çıkmaktadır. Buna karşılık, LSTM-AutoEncoder modeli %24 ile %62 arasında değişen daha düşük doğruluk oranları elde etmiştir. Çalışmamız, veri kümesi özelliklerinin model seçimi ve optimizasyonu üzerindeki önemli etkisinin ortaya koymaktadır. Sahte haber tespitinde ML ve DL etkinliğinin kapsamlı bir değerlendirmesini sunan bu araştırma, çevrimiçi yanlış bilgilerle mücadele için daha güçlü stratejilerin geliştirilmesine önemli ölçüde katkıda bulunmaktadır.

1. Giriş

Modern kaynaklar, çoğu İnternet kullanıcısının sosyal ağ siteleriyle aylık olarak etkileşime girdiğini doğruluyor ve bu da sitelerin bireyler ve toplum üzerindeki etkisini doğruluyor. Bu durum, sosyal ağların etkilerini sınırlamak için neyin sahte, neyin gerçek olduğunu tespit etmenin yollarını bulma zorunluluğunu doğurdu. Bu tez, etkisinden kaçınmak için sahte haberleri gerçek haberlerden tespit edebilen bir sistemi ortaya çıkarmayı amaçlamaktadır.

Sahte haber tespiti (FND) üç karmaşık alt kategoriye kapsar: uydurma, aldatmaca ve hiciv. Uydurma, sansasyonel manşetlerin ustalığını ve güvenilir kaynaklardan gelen bilgilerin kasıtlı olarak çıkarılmasını gerektirir ve çarpık bir anlatı yaratır. Aldatmacalar, halk arasındaki güvenilirliği artırmak için birden fazla yayılma kanalından yararlanarak daha karmaşık aldatma teknikleri kullanır. Öte yandan hiciv, mizah kisvesi altında hayali bilgiler

sunar; bu, yanlış anlaşılırsa gerçekmiş gibi yanlış yorumlanabilecek incelikli bir yanlış bilgi biçimidir ve gerçeği aldatmadan ayırma manzarasını daha da karmaşık hale getirir.

Sahte haberleri tespit edebilecek bir sistem oluşturabilmek için birden fazla veri setine birden fazla algoritma uygulanması gerekti. İkisi mevcut veri seti, ikisi ise birden fazla verinin birleştirilmesiyle oluşturulan dört veri seti kullanıldı. Daha sonra tüm sonuçlar karşılaştırıldı. Bazıları veri setinin sonuçlara etkisini belirlemek içindi. İster makine öğrenmesinde ister derin öğrenmede birden fazla algoritmanın kullanılmasının yanı sıra birden fazla algoritmanın kullanılması veri seti, modeli haber tespitinde daha kapsamlı hale getirme etkisine sahipti.

Yanlış haber yaymanın sonuçları, halkın güvenini ortadan kaldırmaktan şiddeti kışkırtmaya kadar geniş kapsamlı ve ciddidir. Bilginin bütünlüğünü korumak ve halkın güvenilir ve doğru kaynaklara erişimini sağlamak için sahte haberleri etkili bir şekilde ayırt edip vurgulayabilen kesin metodolojiler geliştirmek zorunlu hale geliyor.

Bu çalışma, yanlış haber tespiti için en etkili teknikleri belirlemeyi amaçlamaktadır. Araştırma, her modelin yalan haber tespit etme performansını titizlikle analiz ederek, yalan haber tespitine yönelik daha dayanıklı ve etkili araç ve metodolojilerin oluşturulmasına katkıda bulunmaya çalışıyor. Bu gelişmeler, yanlış bilgi ve dezenformasyona ilişkin etkilerin azaltılmasında önemli bir rol oynamakta ve böylece kamuya sunulan bilgilerin kalitesini ve güvenilirliğini artırmaktadır.

Yapay zekanın (AI) son zamanlardaki evrimi, yalan haber ikileminin üstesinden gelmek için umut verici yollar sunuyor. Yapay zeka teknolojileri, çeşitli kaynaklardan gelen kapsamlı verileri analiz ederek yanlış bilgilerin yayıldığını gösteren davranış kalıplarını belirleyebilir. Bu teknolojiler, yanıltıcı veya yanlış içeriğin tespit edilmesinde ve işaretlenmesinde önemli bir rol oynuyor ve böylece yanlış haberlerin olumsuz etkisini azaltıyor. Yapay zekanın önemli bir bileşeni olan Doğal Dil İşleme (NLP), insan dilinin analiz için makineler tarafından anlaşılabilir bir şekilde işlenmesini ve yorumlanmasını sağlar. Yanlış haber tespitine ML modellerinin ve DL tekniklerinin uygulanması, yanlış haberleri belirlemek ve bunlara karşı koymak için etkili yöntemler geliştirmek için bir yol sağlar.

Bu araştırmanın temel motivasyonu, sahte haberlerle ilgili kritik zorluklara karşı yapay zeka, NLP, makine öğrenimi ve DL'nin potansiyelinden yararlanmaktır. Bu tez, yanlış haber tespitine yeni algoritmalar ve modeller sunmayı, yanlış bilgileri etkili ve hızlı bir şekilde tanımlama ve bunlara karşı koyma yeteneğini artırmayı amaçlamaktadır. Yapay zeka teknolojisinin bu çabada kullanılması, yanlış haberlerin tespitini dönüştürme potansiyeline sahiptir; bilgi dağıtımının doğruluğunu korumak ve insanları ve toplulukları yanlış bilgilerin zararlı etkilerinden korumak için önemli araçlar sunar.

Bu tez öncelikle, insanları ve toplumu ciddi risk altına sokan, artan yanlış haber tehdidiyle acilen mücadele etme ihtiyacından hareket etmektedir. Bu proje yapay zeka, NLP, ML ve DL tekniklerini kullanarak yanlış haber tespiti alanını geliştirmeyi amaçlıyor. Yanıltıcı bilgilerin tespit edilmesi ve çürütülmesi için etkili ve otomatikleştirilmiş stratejiler

geliştirmeye odaklanır. Nihai hedef, güvenilir bilgilere dayanarak kararlar alabilen, daha dirençli ve bilgili bir toplum yetiştirerek yanlış haberlerin olumsuz sonuçlarını azaltmaktır.

Son olarak bu çalışma, kayda değer yanlış haberlerin belirlenmesinde metin analizi ve kategorizasyona yönelik yeni yaklaşımlar geliştirerek NLP ve ML disiplinlerini ilerletmektedir. Yanlış haber tespitiyle etkili bir şekilde mücadele etmek için güvenilir ve aldatıcı bilgi kaynakları arasında ayırım yapacak yeni yaklaşımlara ihtiyaç vardır. Bu tezde sunulan DL algoritmaları, metin analizi ve sınıflandırmaya yeni bakış açıları sunarak NLP ve ML'de kullanılan yöntemleri geliştirir. Bu çalışma, halihazırda var olan bilgi birikimine katkıda bulunmanın yanı sıra, bu alanlarda gelecekteki araştırmalara ve daha gelişmiş algoritmalar ve modellerin geliştirilmesine de kapı açmaktadır.

2. Materyal ve Metot

Sahte haberleri tespit edebilecek bir sistem oluşturabilmek için birden fazla veri setine birden fazla algoritma uygulanması gerektiği. İki set mevcut veri seti, ikisi ise birden fazla verinin birleştirilmesiyle oluşturulan dört veri seti kullanıldı. Daha sonra tüm sonuçlar karşılaştırıldı. Bazıları veri setinin sonuçlara etkisini belirlemek içindi. İster makine öğrenmesinde ister derin öğrenmede birden fazla algoritmanın kullanılmasının yanı sıra birden fazla algoritmanın kullanılması veri seti, modeli haber tespitinde daha kapsamlı hale getirme etkisine sahiptir.

Metodolojimiz, her biri FND'nin bütünsel bir anlayışına katkıda bulunan birkaç temel aşamadan oluşmaktadır. Süreç, uygun bir veri kümesinin seçilmesiyle başlar ve bunun sonraki araştırma aşamaları için temel oluşturma rolünün önemi vurgulanır. Daha sonra veri ön işlemenin önemini inceleyerek ham veri kümesinin nasıl temizlendiğini, iyileştirildiğini ve dönüştürüldüğünü açıklayarak pratik model eğitime hazırlıyoruz.

Çalışmamız öncelikle DL modellerinin uygulanmasıyla ilgilidir. CNN, CNN-LSTM ve CNN-GRU mimari kombinasyonlarının mimarisini ve seçim kriterlerini tartışıyoruz. Kapsamlı ve güvenilir bir analizin garanti edilmesi için katmanların seçimine, filtre boyutuna ve aktivasyon fonksiyonlarına özellikle dikkat edilir. Daha sonra test ve eğitim prosedürleri, modellerimizin verimliliğini ve geniş çapta uygulanabilirliğini garanti etmek için sektördeki en iyi uygulamalar kullanılarak ele alınır.

Çalışmamız, veri setlerini birleştirerek çok boyutlu bir yaklaşımı birleştirmesi nedeniyle önceki araştırmalardan farklılık göstermektedir. Bu kombinasyon, "kısmen doğru" ve "biraz doğru" gibi daha ayrıntılı kategorileri içerecek şekilde temel ikili sınıflandırmanın ötesine geçen dikkatli incelemeyi kolaylaştırır. Bu katmanlı ve hiyerarşik veri seti, sınıflandırma yönteminin karmaşıklığını ve derinliğini artırmak için tasarlanmıştır.

FND'de birkaç ardışık aşamadan oluşan yapılandırılmış süreç söz konusudur. İlk adım, özellikle bu alanla ilgisiyle tanınan ISOT veri kümesini kullanarak veri toplamayı içerir. Veri toplamanın ardından kapsamlı bir ön işleme aşaması gerçekleştirilir. Bu aşama, sonraki aşamalara hazır olmasını sağlamak için verilerin temizlenmesine ve hassaslaştırılmasına odaklanır.

Ön işleme sonrasında odak noktası, önerdiğimiz sınıflandırma modellerini uygulamaya geçer. Karmaşık bir şekilde tasarlanmış bu modeller, sahte haberlerin belirlenmesinde çok önemlidir. Kurulumlarının ardından bu sınıflandırıcılar, işlenmiş veri kümesinden bilgiyi özümstedikleri bir eğitim aşamasına girer. Bu eğitim, modellerin parametrelerine ince ayar yapmak ve doğru tahmin yapma yeteneklerini artırmak için gereklidir.

Eğitim tamamlandıktan sonra modeller, doğrulama için aynı hassas veri kümesinin farklı alt kümelerini kullanan bir test aşamasından geçer. Bu adım, genelleme becerilerini ve genel performanslarını değerlendirmek için kritik öneme sahiptir. Son aşama, her modelin etkinliğini değerlendirmek için derinlemesine bir analiz içerir. Bu değerlendirme, daha önce de belirtildiği gibi, her modelin güçlü yönlerine ve sınırlamalarına ilişkin kapsamlı bir fikir veren bir dizi performans ölçümünü içerir. Ayrıca, farklı modeller arasındaki performans farklılıklarını ayırt etmek için karşılaştırmalı analizler yapılmaktadır.

Sahte haberleri etkili bir şekilde tanımlamak ve kategorize etmek için haber raporlamada doğruluğun çeşitli yönlerini kapsayan kapsamlı bir veri kümesine sahip olmak zorunludur. Sonuç olarak, metodolojimiz dört farklı veri kümesini birleştirerek daha ayrıntılı ve kapsamlı bir incelemeyi kolaylaştırır.

ISOT Corpus

Analizimizde, haber makalelerinin orijinal kaynaklardan türetilen "sahte" veya "gerçek" kategorilere göre ikili olarak sınıflandırılması için özenle bir araya getirilmiş bir veri kümesi olan ISOT külliyyatını kullanıyoruz. Bu derleme, 21.417'si gerçek ve 23.481'i sahte olarak sınıflandırılmış ve her biri 12.600'den fazla makale içeren "True.csv" ve "Fake.csv" olmak üzere iki ayrı dosyaya bölünmüş 44.898 makale içermektedir.

FakeNewsNet Veri Kümesi

Sahte haberlerin inceliklerini araştıran akademisyenler için oluşturulmuş özel bir kaynak, FakeNewsNet veri kümesidir. Politifact ve Gossipcop olmak üzere iki farklı ve güvenilir kaynaktan alınan doğrulanmış gerçekler üzerine kuruludur ve sahte haberlerin inceliklerini incelemek için kapsamlı bir bakış açısı sunar.

23.196 satır ve beş temel sütundan oluşan FakeNewsNet veri seti, sahte haberlerin yayılmasında yer alan dinamiklerin kapsamlı bir şekilde anlaşılmasını sağlayacak şekilde düzenlenmiştir. Tasarımı, beş temel özelliği bir araya getirerek araştırmacının erişilebilirliği ve verimliliği için özel olarak tasarlanmıştır: Kimlik, Haber_URL'si, Başlık, Tweet_ID'ler, Etiket

FakeNewsNet veri kümesi, gelişen FND ve anlayış alanındaki karmaşık analizlere uygun, iyi organize edilmiş ve zengin bir kaynaktır.

LIAR-PLUS Veri Kümesi

LIAR-PLUS veri kümesindeki öğelerin dahil edilmesi, FND ve sınıflandırma metodolojimizi önemli ölçüde geliştirir. Doğruluk, konu ve bağlam gibi standartlara göre

manuel olarak açıklama eklenmiş 12.836 kısa ifadeyi içeren bu veri kümesi, ilk kez 2017 yılında Wang tarafından sunulan orijinal LIAR veri kümesinin bir uzantısıdır. Veri kümesinde doğruluk, şu şekilde kategorize edilmiştir: "pantolon ateşi"nden "gerçek"e kadar bir spektrum.

LIAR-PLUS veri seti, Politifact gazetecileri tarafından yazılan tam metin karar raporlarından otomatik olarak çıkarılan gerekçeler veya delil cümleleri eklenerek önemli ölçüde iyileştirildi. Genellikle "kararımız" veya "özetleme" gibi başlıkların yer aldığı özet bölümlerinde bulunan bu açıklamalar, atanan dürüstlük notunun bağlamını sağlar. Özet bölümü yoksa, sonucu açıkça belirten herhangi bir cümle yerine, doğrulama makalesinin son beş cümlesi kullanılır.

LIAR-PLUS veri seti, kapsamlı ve çok boyutlu bir bakış açısı sunan 15 sütundan oluşur: Kimlik, Etiket, Konular, Konuşmacı Detayları, Durum Bilgileri, Parti Bağlantısı, Konuşmacının kredi geçmişi, önceki doğruluk derecelendirmelerine göre değerlendirilir, bunlar "neredeyse doğru", "yanlış", "yarı doğru", "çoğunlukla doğru" ve "pantolon yanıyor" olarak kategorize edilmiştir., Bağlam, Alıntılanan Gerekçe.

LIAR-PLUS veri setini entegre etmedeki amacımız, FND algoritmalarımızın çok daha iyi performans göstermesine yardımcı olacak kanıt alma kalitesini artırmaktır. Haber parçalarının bütünlüğünü belirlemek için gerekli bağlamsal derinliği sağlamak amacıyla bu gerekçe cümlelerini kullanarak sınıflandırma sürecini güçlendirmeyi umuyoruz.

Veri kümelerinin birleştirilmesi

ISOT külliyyatının LIAR-PLUS veri seti ile dikkatli bir şekilde entegrasyonu sayesinde, FND hakkındaki araştırmamız önemli bir gelişmeyi temsil etmektedir. Basit ikili sınıflandırmanın ötesinde, bu füzyon, "Yarı Doğru" ve "Zar zor Doğru" gibi kategorileri birleştirerek FND'ye daha incelikli bir yaklaşıma olanak tanır. Daha sonra ISOT külliyyatını LIAR-PLUS'ın "Yarı Doğru" kategorisiyle birleştirerek Veri Kümesi 1 olarak etiketlediğimiz ilk veri kümemizi oluştururuz. Bunun üzerine inşa etmek için Veri Kümesi 2 adı verilen ve "Zar zor Doğru" kategorisini içeren başka bir veri kümemiz var. " kategorisini kullanır ve Veri Kümesi 1'i büyütür. ISOT derleminin LIAR-PLUS'un "Yarı Doğru" ve "Zar zor Doğru" bileşenleriyle birleştirilmesiyle bileşik bir veri kümesi oluşturulur.

Ön işleme

ML modellerinin etkinliği, girdi verilerinin kalitesinden büyük ölçüde etkilenir ve verileri pratik analiz için optimize etmek amacıyla kapsamlı ön işleme adımları gerektirir. Ön işleme hattımız, her biri modelin anahtar terimleri tanımlama kapasitesini geliştirmeyi ve dolayısıyla performansını artırmayı amaçlayan bir dizi teknik içerir. Ön işleme rejimimizdeki adımlar şunları içerir:

Öznitelik Seçimi

Ana veri kaynaklarımız olan ISOT derlemindeki ve LIAR-PLUS'taki "başlık" ve "metin" özelliklerine odaklanarak başlıyoruz. FakeNewsNet veri kümesi için dağıtım dengesini,

"gerçek" haber makalelerini "sahte" haber hacmiyle eşleştirecek şekilde alt örnekleyerek elde ediyoruz. Her veri kümesi için birincil özellik setimiz, özenle seçilmiş bu iki özelliğin (başlık ve metin) tek bir sütunda birleştirilmesiyle oluşturulur.

Metin Temizleme

Bir sonraki adım, tüm metni küçük harfe çevirmeyi ve HTML etiketlerini, noktalama işaretlerini ve gereksiz boşlukları ortadan kaldırmak için normal ifadeler kullanmayı içerir. Bu standardizasyon, veri kümesi genelinde tutarlılığı garanti eder ve basitleştirmeye yardımcı olur.

Tokenizasyon

Doğal Dil İşleme'de (NLP) önemli bir teknik olan tokenizasyon üçüncü adımdır. Temizlenen metni kelimeler veya kelime öbekleri gibi daha küçük birimlere veya belirteçlere ayırır. Bu süreç, metni basitleştirerek temel öğelerin bir listesine dönüştürür ve özellik çıkarımı için zemin hazırlar.

Dönem Frekansı-Ters Belge Frekansı (TF-IDF)

Bir derlem hakkındaki bir belgedeki bir kelimenin önemini değerlendirmek için istatistiksel bir teknik olan TF-IDF'yi kullanıyoruz. Belirli bir belgede önemli olan ancak bütünde yaygın olmayan sözcükler, terim sıklığının (TF) ters belge sıklığıyla (IDF) çarpılmasıyla belirlenen TF-IDF puanıyla vurgulanır. Bu onları değerli farklılaştırıcılar yapar.

Veri Bölme

Kapsamlı ön işlemenin ardından veri kümesi, akademik araştırmamızda tahmin modelimizi eğitmek ve test etmek için akıllıca çeşitli alt kümelere bölündü. Veri setinin %80'ini eğitim için, kalan %20'sini ise test seti olarak özel olarak seçtik. Bu bölüm, test aşamasında modelin performansının tarafsız bir değerlendirmesine olanak tanırken, eğitim aşaması için güçlü bir örnek boyutu sağlayacak şekilde titizlikle tasarlandı. Ek olarak, model değerlendirmemizin güvenilirliğini artırmak için beş kat kullanan katmanlı bir K-Katlama çapraz doğrulama yaklaşımı kullandık. Bu yöntem, her tarihin modeli değerlendirmesine izin vererek eğitim derleminin değerini artırır ve her katlamada sınıfların orantılılığını korur. Bu stratejilerin birleştirilmesi, modelimizin dijital haber alanını etkileyen çeşitli ve karmaşık yanlış bilgi ifadelerini daha büyük bir doğruluk ve hassasiyetle tanımasına ve değerlendirmesine olanak tanır.

Makine öğrenimi tabanlı sınıflandırıcılar

ISOT veri setindeki sahte haberlerin verimli bir şekilde tespit edilmesi için çalışmamızda sekiz farklı ML algoritması uyguladık. Burada her algoritmayı inceleyerek temel fikirlerini ve yeteneklerini vurguluyoruz.

Karar Ağacı (DT)

DT algoritması, sınıflandırma ve regresyon görevlerindeki basit yaklaşımıyla tanınan, ML'de bir temel taşıdır [83]. Özünde, bir DT, bir veri kümesini giderek daha ince alt kümelerle bölerken aynı anda ilişkili bir DT'yi oluşturur. Nihai model, karar ve yaprak düğümlerinden oluşan bir ağaçtır. Karar düğümleri bir giriş özelliği için farklı olası değerleri temsil ederken, yaprak düğümler tüm girdiler dikkate alındıktan sonra nihai kararı veya çıktıyı belirtir.

Maliyet-karmaşıklık budaması gibi budama teknikleri, daha az etkili dalları ortadan kaldırarak aşırı uyumu önlemek için kullanılır. Bu, ağaç boyutunu ve uyumunu dengelemek için bir 'alfa' parametresini entegre eder. Ek olarak, ağaç derinliği, yaprak boyutu ve bölme başına özellik sayısı gibi hiper parametrelere ince ayar yapılarak yetersiz uyum ve aşırı uyum dengelenir.

Rastgele Orman (RF)

RF tekniği, EL'in temel bir bileşenidir ve bu nitelikler için iyi bilinen bir model oluşturmak üzere çok sayıda DT'nin birleşik gücünden ve sağlamlığından yararlanır. Yaklaşım, birden fazla zayıf öğrenicinin tek bir güçlü öğrenici oluşturmak için birleştirilebileceğini ve modelin tahmin gücünü ve istikrarını geliştirebileceğini öne süren EL teorisine dayanmaktadır. Merkezi limit teoremi ve büyük sayılar kanunu bu fikri destekleyen iki temel matematiksel ve olasılıksal fikirdir. Bu fikirlere göre, oldukça fazla sayıda nispeten bağımsız ağacın birleştirilmesi, normal bir dağılıma doğru yaklaşarak tahmin doğruluğunu artırır ve sonuçları stabilize eder.

RF algoritmasının eğitim aşaması sırasında, torbalamaya dayalı bir yaklaşım olan hem özellikleri hem de veri noktalarını seçen stokastik bir süreç aracılığıyla çok sayıda DT geliştirilir. Topluluktaki her ağaç, kombinatorial ve optimizasyon tekniklerine dayanan bir yöntem olan, her düğümdeki özelliklerin rastgele alt kümelerinin seçilmesiyle sağlanan ağaçlar arasındaki çeşitlilikle bağımsız olarak yeni verileri tahmin eder. Bireysel ağaçları eğitmek için farklı önyükleme örnekleri üreten torbalama işlemi, önyargıyı biraz artırırken tahmin varyansını azaltmayı amaçlamaktadır. RF'deki son tahmin, ya regresyon için tüm ağaç tahminlerinin ortalamasıdır ya da sınıflandırma için çoğunluk oyu olup, tipik olarak gelişmiş genelleme hatasıyla sonuçlanır. Bu hata, model karmaşıklığı ile tahmin doğruluğu arasındaki dengeyi anlamak için çok önemli olan önyargı-varyans ayrıştırması kullanılarak incelenir.

Gradyan Arttırma

Daha zayıf öğrenme modellerini, özellikle DT'yi metodik olarak birleştiren GB algoritması, matematiksel ve hesaplamalı kavramlara dayanan gelişmiş bir EL tekniğidir ve

Gradyan iniş yöntemi, sistemi tekrar tekrar kayıp fonksiyonunu en aza indirmeye yönlendirmek için kullanılır; serideki her ağaç, öncekilerin hatalarını bu organize şekilde iyileştirmeye çalışır. Her ağacın katkısını etkileyen öğrenme oranı gibi bir dizi faktör, aşırı

uyumu önleyen ve daha kapsamlı modellerin oluşturulmasını teşvik eden kontrol mekanizmaları olarak hizmet eder.

GB tekniği, olağanüstü derecede kesin ve doğru tahmin modelleri oluşturmak için işbirliği yapan istatistik, optimizasyon ve hesaplama ile ilgili fikirlere dayanan karmaşık bir araçtır. Çeşitli tahmin senaryolarında olağanüstü işlev alanı optimizasyonu ve karmaşıklık uyarlaması sergileyen çağdaş ML'nin önemli bir bileşenidir. Bu, onu günümüzün makine öğrenimi araç kutusunda, karmaşık tahmine dayalı modelleme zorluklarını doğru ve kesin bir şekilde ele alabilen önemli bir araç haline getiriyor.

Lojistik Regresyon (LR)

Hem makine öğreniminin hem de istatistiğin önemli bir bileşeni olan LR, esnek ve temel bir denetimli öğrenme tekniğidir.

Bu fonksiyon girdi olarak gerçek bir sayıyı kabul eder ve 0 ile 1 arasında bir değer döndürür; bu, belirli bir girdinin belirli bir kategoriye girme olasılığını gösterir.

Sırt ve kement düzenlemesi gibi düzenleme stratejileri, gerçek dünyadaki LR uygulamalarında aşırı uyumu önlemek için sıklıkla kullanılır. Bu teknikler, her özelliğe atanan ağırlıkların büyüklüğünü sınırlayarak yeni, görünmeyen verilere genelleme yapmada iyi performans gösteren daha basit modelleri destekler. Ayrıca, titiz özellik mühendisliği ve seçimi, LR modellerinin etkinliği açısından çok önemlidir. Ek olarak LR, hesaplama açısından kolayca anlaşılabilen bir mimariye sahiptir; bu da, modelin yorumlanabilirliğinin tahmin doğruluğundan öncelikli olduğu durumlarda onu iyi bir seçenek haline getirir. Basit olmasına rağmen LR, özellikle modele doğrusal olmama özelliği eklemek ve tahmin gücünü artırmak için polinom regresyonu gibi diğer yöntemlerle eşleştirildiğinde etkili bir araçtır.

XGBoost

XGBoost, olağanüstü hız ve verimlilik için tasarlanmış gelişmiş bir GB DT uygulamasıdır. Doğru tahminler yapabilen güçlü bir öğrenci yetiştirmek için bu yöntem, genellikle DT'ler şeklinde bir grup zayıf öğrenciyi bir araya getirir. Topluluğun önceki üyelerinin yaptığı hataları düzelter zayıf öğrenenler ekleyerek türevlenebilir bir kayıp fonksiyonunu optimize etmeye dayanan GB tekniğini kullanır.

Bir düzenleme terimi ve belirli bir kayıp fonksiyonundan oluşan XGBoost'un amaç fonksiyonu, algoritmanın matematiksel temelini açıklamak için kullanılabilir. Algoritma, beklenen ve gerçek değerler arasındaki farkı ölçen kayıp fonksiyonu aracılığıyla en doğru tahminleri üretmeye yöneliktir. Öte yandan düzenleme terimi, modelin karmaşıklığını düzenleyerek aşırı uyumun önlenmesine yardımcı olur.

XGBoost algoritması, gelişmiş optimizasyon yöntemlerini etkili düzenleme ve hesaplama taktikleriyle birleştirerek olağanüstü tahmin performansı ve doğruluk sunan, makine öğrenimi profesyonelleri için güçlü bir araçtır. Sağlam bir matematiksel temele dayanmaktadır.

Çok terimli Naif Bayes

Multinomial Naïve Bayes (MNB), NLP'de yaygın olarak kullanılan olasılıksal bir öğrenme yöntemidir. NB sınıflandırıcısının bir çeşidi olarak MNB, sınıf değişkeninin değeri verilen her özellik çifti arasındaki "saf" koşullu bağımsızlık varsayımına sahip Bayes teoremine dayanmaktadır. Bu model, özelliklerin sayımlar veya sıklık sayımları olarak temsil edildiği sınıflandırma görevleri için özellikle uygundur, dolayısıyla özelliklerin tipik olarak belgelerdeki sözcük sayıları veya normalleştirilmiş sözcük sıklıkları olduğu metin sınıflandırma problemlerine iyi bir şekilde katkıda bulunur.

MNB denklemindeki payda, olasılıkların 1'e kadar toplanarak normalleştirilmesini sağlar. Yeni bir belgenin sınıflandırılması gerektiğinde, MNB algoritması her sınıf için sonsal olasılığı hesaplar ve belgeyi en yüksek sonsal olasılığa sahip sınıfa atar.

Sadeliğine ve bazen aşırı güçlü bağımsızlık varsayımına rağmen MNB, özellikle özelliklerin çok terimli olarak dağıtıldığı alanlarda karmaşık görevlerde şaşırtıcı derecede iyi performans gösterir ve bu da onu metin sınıflandırma araç setinde temel bir algoritma haline getirir.

Destek Vektör Makinesi (SVM)

SVM, problemler ve aykırı değerlerin tespiti için popüler bir denetimli makine öğrenimi tekniğidir. SVM'nin arkasında yatan fikir, veri noktalarını belirli bir özellik alanı içinde farklı sınıflara verimli bir şekilde bölmek için ideal hiperdüzlemi bulmaktır. Bu hiperdüzlem, ikili sınıflandırma senaryolarında farklı sınıflara ait destek vektörleri olarak bilinen en yakın veri noktaları arasındaki marjı maksimuma çıkaracak şekilde konumlandırılır.

SVM, girdi özelliklerini daha yüksek boyutlu bir uzaya eşleyen bir çekirdek tekniği kullanır; burada bir hiperdüzlem, uygulamasını doğrusal olarak ayrılabilir konuların ötesine taşımak için sınıfları başarılı bir şekilde ayrılabilir. RBF, polinom ve doğrusal çekirdekler sıklıkla kullanılan çekirdeklerdir. Çekirdek seçimi ve parametre ayarı, makinenin performansını büyük ölçüde etkilediğinden SVM'nin dağıtımında çok önemli aşamalardır.

Matematikteki esnekliklerine ve titizliklerine rağmen, SVM'lerin eğitim sırasında ikinci dereceden programlama sorununun hesaplama maliyeti nedeniyle özellikle büyük veri kümeleri için ölçeklenebilirlik konusunda sınırlamaları vardır. Ancak SVM, doğru optimizasyonlar ve çekirdek seçimleri verildiğinde, çeşitli veri ortamlarında önemli modelleri ortaya çıkarabilen ve karmaşık sınırlar çizebilen, makine öğrenimi araç kutusunda güçlü bir araç olmaya devam ediyor. Bu, endüstriyel ve bilimsel kullanımlar açısından önemini vurgulamaktadır.

Derin öğrenmeye dayalı sınıflandırıcılar

İncelenen zorluklar karmaşık olduğundan, çeşitli model konfigürasyonlarını dikkatli bir şekilde oluşturuyoruz. Bu çalışmada, iki temel mimari paradigmaya odaklanarak DL'nin gelişmiş yeteneklerini kapsamlı bir şekilde araştırıyoruz: CNN'ler ve RNN'ler. Bunlar, CNN-GRU ve CNN-LSTM hibrit mimarilerini ve bağımsız CNN ve RNN modellerini içerir. Her

model, spesifik bilimsel soruları ve hesaplama zorluklarını özel olarak ele almak için yapılmıştır.

Evrişimli Sinir Ağı (CNN)

Görsel görüntüleme işleme için kullanılan popüler bir DNN türü CNN'dir. Girdi verilerinden hiyerarşik mekansal özelliklerin otomatik ve uyarlanabilir şekilde öğrenilmesi, mimarisinin temelini oluşturur. Girdi verilerindeki kalıpları tanımlayan nöron katmanlarının giderek daha karmaşık hale gelmesiyle, CNN'ler görsel korteksin bilgi işlemesini taklit edecek şekilde tasarlanmıştır. Daha yüksek seviyeli özellikler, standart bir CNN mimarisinde ayrı katmanların bir yığını tarafından giriş verilerinden kademeli olarak çıkarılır.

Giriş sırası sözcüklerini 128 birimlik yüksek boyutlu vektörlere dönüştüren bir gömme katmanı ile başlayarak bu modelle sırayla çalışıyoruz. Daha yüksek seviyeli işlemler için bu, girişin yoğun bir temsilini sunar. Bırakma katmanları, az miktarda veriyle eğitilen birçok modelin başına gelen aşırı uyumu durdurmak için kullanılır.

Daha sonra, verilerdeki yerelleştirilmiş modeller, her biri 128 filtreye sahip üç farklı evrişim katmanı tarafından yakalanır. Çekirdek boyutlarının (3, 4 ve 5) değiştirilmesiyle geniş bir aralıkta uzamsal özellik çıkarımı sağlanır. Modelin dahili temsilinin boyutluluğunu azaltarak, küresel maksimum havuzlama katmanı sınıflandırma için gerekli özellikleri korur. Sonuçta, bu özellikler bir veya daha fazla yoğun katman tarafından karmaşık korelasyonların keşfedilebileceği bir alana haritalanır. Modelin sonunda bir sigmoid fonksiyonu tarafından etkinleştirilen yoğun bir katman, her sınıf etiketi için sınıflandırma olasılıkları üretir.

LSTM

LSTM ağları, geleneksel RNN'lerin eğitiminde karşılaşılan zorlukların üstesinden gelmek için titizlikle hazırlanmış özel bir Tekrarlayan Sinir Ağı (RNN) mimarisidir. Bu zorluklar öncelikle, diziler içindeki uzun vadeli bağımlılıkları yakalama ve öğrenme yeteneğini engelleyen, kaybolan ve patlayan gradyan sorunlarından kaynaklanmaktadır. LSTM'ler, bilgi akışını kontrol eden bir dizi geçit mekanizmasını birleştirerek bu engellerin üstesinden gelmek için özel olarak tasarlanmıştır.

Bir LSTM ünitesinin mimari karmaşıklığı üç farklı kapıyla karakterize edilir: giriş kapısı, unutma kapısı ve çıkış kapısı. Bir LSTM hücresindeki her kapı, sırasıyla yeni bir girdinin sahip olması gereken etki derecesini, mevcut bilginin ne ölçüde korunması gerektiğini ve bir sonraki zaman adımına aktarılacak iç durumların oranını belirleyen bir nöbetçi gibi çalışır.

LSTM'lerin pratik sonuçları derindir; çünkü dil modelleme, çeviri, dizi tahmini ve daha fazlasını içeren çok çeşitli görevlerde mükemmeldirler. LSTM içindeki kapıların ve durumların karmaşık etkileşimi, bilgilerin uzun süreler boyunca hatırlanmasına ve ilgisiz verilerin seçici olarak unutulmasına olanak tanır. Sonuç olarak, LSTM'ler sinir ağları alanında önemli bir ilerlemeyi temsil eder ve daha önce makine anlayışının ötesinde olduğu düşünülen sıralı verilerdeki kalıpların keşfedilmesine ve öğrenilmesine olanak tanır.

Otomatik Kodlayıcı

Otomatik kodlayıcılar, boyutluluk azaltma ve özellik öğrenme arayışını somutlaştıran bir sinir ağı paradigmasını temsil eder. Otomatik kodlayıcının temel amacı, orijinal girdinin sıkıştırılmış bilgi temsilini zorlayan ağ içinde bir darboğaz oluşturarak, tipik olarak boyut azaltma amacıyla belirli bir veri kümesi için bir temsili (kodlama) öğrenmektir.

Bir otomatik kodlayıcının mimarisi, iki ana bileşenden oluşan zarif bir simetri sergiler: kodlayıcı ve kod çözücü. Kodlayıcının rolü, giriş verilerini gizli alan temsiline sıkıştırmaktır. Bunu, giriş verilerinin boyutlarını çeşitli katmanlar aracılığıyla kademeli olarak azaltarak ve sonuçta kodlanmış (sıkıştırılmış) bilgileri içeren darboğaza ulaşarak başarır. Buna karşılık, kodlayıcıyı yansıtan kod çözücü, bu kodlanmış girişi alır ve verileri orijinal boyutluluğuna göre yeniden yapılandırır.

Otomatik kodlayıcıların mimari yapısı ve eğitim yaklaşımı, onları son derece çok yönlü hale getirir; veri gürültüsünü gidermeye, üretken modellemeye ve hatta diğer sinir ağları için bir ön eğitim adımı olmaya uygundur. Veri odaklı çağda ilerledikçe, otomatik kodlayıcıların önemi artmaya devam ediyor ve karmaşık veri kümelerinin içinde gizlenen doğal yapılara ve kalıplara ilişkin değerli bilgiler sağlıyor.

LSTM-Otomatik Kodlayıcı

Gelişmiş sinir ağı araştırmalarında, hibrit LSTM-Otomatik Kodlayıcı mimarisi, hem LSTM hem de Otomatik Kodlayıcıların güçlü yönlerinden yararlanan gelişmiş bir model olarak ortaya çıkmıştır. Bu kompozit yapı, zamansal bağımlılıkları yakalamak ve boyutsallık azaltmayı gerçekleştirmek için ustalıkla tasarlanmıştır; özellikle zaman serisi analizi, sinyal işleme ve doğal dil işleme gibi alanlarda yaygın olan dizi verilerinin işlenmesi için çok uygundur.

Hibrit model kusursuz bir şekilde entegre edilmiş iki bileşenden oluşur: LSTM katmanları ve Otomatik Kodlayıcı çerçevesi. Başlangıçta, LSTM katmanları kodlayıcı görevi görür ve giriş dizisi verilerini tekrarlayan bağlantıları ve bilgi akışını kontrol eden kapılı mekanizmalar aracılığıyla işleyerek zamansal dinamikleri ustaca yakalar. LSTM'nin bilgiyi seçici olarak hatırlama ve unutma konusundaki benzersiz yeteneği, diziyi darboğazındaki sabit boyutlu bir vektöre kodlamasına ve girdi dizisinin zamansal özelliklerini kapsüllemesine olanak tanır.

Daha sonra bu vektör, kodlanmış gösterim görevi görür ve bu daha sonra mimarinin Otomatik Kodlayıcı bölümüne girilir. Otomatik Kodlayıcı tipik olarak kodlayıcının mimarisini yansıtan ancak bunun tersini yapan bir kod çözücüye sahiptir. Amacı, LSTM'nin kodlanmış vektöründen orijinal diziyi yeniden oluşturmaktır. Yeniden yapılandırma aşaması, ağı, temel zamansal özelliklerini koruyan dizi verilerinin sıkıştırılmış bir temsilini öğrenmeye zorladığı için çok önemlidir.

Hibrit bir LSTM-Otokodlayıcının eğitim süreci, zaman içinde geri yayılımı (BPTT) içerir; bu sırada model, orijinal giriş dizisi ile kod çözücü tarafından yeniden oluşturulan dizi çıkışı arasındaki farklılığı en aza indirmeyi öğrenir. Bu yinelemeli süreç, LSTM'nin önemli

zamansal özellikleri yakalamasını teşvik ederken, Otomatik Kodlayıcının yeniden yapılandırma yeteneklerini geliştirmesine olanak tanır. Sonuç, sıralı verilerin sıkıştırılmış ancak bilgilendirici bir temsildir.

Hibrit LSTM-Otomatik Kodlayıcı modeli, karmaşık sıralı bilgileri ayrıştırma ve yorumlama kapasitemizde önemli bir ilerlemeyi temsil eder ve DL alanındaki en zorlu zorluklardan bazılarını zarif bir çözüm sunar.

CNN-LSTM Hibrit

Bu CNN-LSTM hibrit yaklaşımı, video sınıflandırma, NLP ve zaman serisi analizi gibi alanlarda yaygın olarak karşılaşılan karmaşık zorlukları çözmek için her iki mimarinin ilgili güçlü yönlerinden yararlanır. Bu alanlar genellikle önemli mekansal ve zamansal dinamiklere sahip verileri içerir ve bu da mekansal ve sıralı bilgilerin entegre edilmesini çok önemli hale getirir.

Giriş verilerinden mekansal hiyerarşiler elde etmeye odaklanan bu hibrit mimarideki CNN bileşeni, hiyerarşik bir özellik çıkarıcı olarak işlev görür. Bir dizi evrişimli katman ve havuzlama katmanı, genellikle uzamsal boyutları en aza indirmek ve kritik bilgileri korumak için kullanılır. Bu aşama, havuzlama, evrişim ve doğrusal olmayan aktivasyon gibi teknikleri kullanarak mekansal özelliklerin çıkarılmasını kolaylaştırır ve boyutluluğu azaltır. Çıktı, girdi verilerini çeşitli düzeylerde temsil eden bir soyutlama katmanıdır.

CNN-LSTM hibrit mimarisinin verimli bir şekilde kullanılmasında önemli bir unsur, hem CNN hem de LSTM bileşenlerinin parametrelerinin ve yapısının kasıtlı olarak düzenlenmesidir. İkisi birlikte, LSTM açısından alınan zamansal olarak akıllı analizleri CNN'den gelen mekansal olarak zengin temsillerle birleştiren bu formda daha etkili bir şekilde çalışır. Nihai sonuç, verilerde mevcut olan zamansal ve mekansal dinamiklerin kapsamlı bir şekilde anlaşılmasıdır.

CNN-LSTM hibrit mimarisinin kullanımı, birçok farklı alanda üstün performansın artırılmasında önemli olmuştur. Bu, zamansal dinamiklerin LSTM aracılığıyla anlaşılmasına ve mekansal bilgilerin CNN kullanılarak eş zamanlı olarak anlaşılmasına olanak tanıyan video analizi gibi alanları içerir. Sıralı verileri dikkatlice inceleyerek mimari, doğal dil işleme için metin verilerinde bulunan yapısal karmaşıklıkları ve bağlantıları yakalamada başarılı olur. CNN-LSTM hibrit modeli, uyarlanabilirliği ve esnekliği nedeniyle zengin mekansal ve zamansal dinamiklere sahip karmaşık veri türlerini yönetmek için etkili bir araçtır.

Yöntemimiz CNN-LSTM mimarisinin avantajlarından yararlanmaktadır. Mimarimiz, giriş dizisindeki her kelimeyi CNN modeline benzer şekilde yoğun bir vektör temsiline dönüştüren bir yerleştirme katmanı ile başlar. Model daha sonra çekirdek boyutu üç ve filtreleri 64 ve 128 olan tek boyutlu evrişim katmanlarını (Conv1D) kullanır. Özellik haritası boyutunun azaltılması yoluyla hesaplama maliyetini en aza indirmek için her evrişim katmanından sonra bir maksimum havuzlama prosedürü uygulanır.

Maksimum havuzlama ve evriřim katmanlarından sonra, önemli bir 64 birimlik LSTM katmanı eklenir. Bu katman, veri dizileri içindeki zamansal bağımlılıkların yakalanmasını mümkün kılar. Aşırı uyumu azaltmak için 0,2 oranında bir bırakma katmanı ekliyoruz. Sonunda, mimariyi tamamlamak için ReLU aktivasyonuna sahip 64 birimlik yoğun bir katman ve ardından ikili sınıflandırma uygulamaları için çok yararlı olan sigmoid fonksiyon aktivasyonuna sahip ikinci, 2 birimlik yoğun bir katman kullanıyoruz.

CNN-GRU Hibrit

Bu gelişmiş hibrit model, zamansal ve uzaysal dinamikler arasında düzgün bir etkileşim gerektiren DL uygulamaları için çok uygundur; çünkü uzaysal modellerin çıkarılmasında ve sıralı bilgilerin yakalanmasında iyi performans gösterecek şekilde dikkatle tasarlanmıştır.

Artan bir filtre hiyerarşisi, bu algoritmanın ilk aşamasını içeren her bir evriřim katmanını belirler. 64 filtreyle başlayıp 128 kat daha büyük bir katmana doğru ilerliyor. Bu evriřimli bileşenler, giriş verilerinden hassas bir şekilde hiyerarşik özellikleri çıkarır. Her bir evriřim aşamasını takip eden maksimum havuzlama katmanlarının eklenmesi bu süreci vurgulamaktadır.

Mimarinin sıralı bileşenine geçildiğinde, 64 birimlik önemli bir GRU katmanı etkinleştirilir. Bu GRU katmanı, benzersiz geçitleme teknikleriyle zamansal bağımlılıkların dikkatli bir şekilde incelenmesini koordine eden sistemin merkezi bileşenidir. Bu süreçler, dizinin her aşamasında hangi verinin saklanması veya yok edilmesi gerektiğine karar vererek bilgi akışını kontrol eder. Mimarinin dayanıklılığını güçlendirmek için, kasıtlı olarak 0,2 oranında bir bırakma katmanı eklenir.

ReLU işlevi, mimari hiyerarşinin sonraki katmanlarının bir parçası olan yoğun bir katmandaki 64 birimi etkinleştirir.

CNN-GRU hibrit modeli, zamansal ve mekansal analizleri düzgün bir şekilde birleşik bir çerçevede birleştiren kapsamlı bir çözümdür. Bu model, mevcut DL projeleri için tahmin doğruluğunu iyileştirmenin potansiyel olarak yararlı bir yolunu sunar.

Değerlendirme önlemleri

Çalışmamızda, uygulanan sınıflandırıcıların değerlendirilmesi, her modelin belirlenen görevi yerine getirmedeki etkinliğinin anlaşılmasında önemli bir rol oynamaktadır. Bu anlayışa ulaşmak için, sınıflandırıcıların etkinliğine ilişkin farklı bakış açıları sunan bir dizi performans ölçütü kullanıyoruz. Bu ölçümler matematiksel denklemlerle titizlikle tanımlanır. Bu ölçümler şunlardır: Geri Çağırma (REC), Hassasiyet (PREC), ACC ve F1-Score (F1).

Bu göstergelerin her biri, bir bütün olarak ele alındığında sınıflandırıcı performansını değerlendirmek için benzersiz bir bakış açısı sağlarken, hem model seçimine hem de gelecekteki araştırma alanlarına rehberlik eden kapsamlı bilgiler sunar.

3. Tartışma

Çalışmamız kendisini FND (FND) alanındaki önceki araştırmalardan birkaç önemli yönüyle ayırmaktadır. İlk olarak, metodolojimiz ISOT külliyatı, FakeNewsNet ve LIAR-PLUS veri seti gibi veri setlerini entegre ederek kapsamlı ve çok boyutlu bir yaklaşımı içermektedir. Bu birleştirme, temel ikili sınıflandırmayı aşarak "Yarı Doğru" ve "Çok Az Doğru" gibi daha incelikli kategorileri içerecek şekilde incelikli bir incelemeyi kolaylaştırır. Bu katmanlı ve hiyerarşik veri derlemesi, sınıflandırma yöntemimizin karmaşıklığını ve derinliğini giderek artırmak ve sahte haberlerin yol açtığı çok yönlü zorlukları daha kapsamlı bir şekilde ele almak üzere tasarlanmıştır.

Ayrıca çalışmamızda RF ve XGBoost'tan CNN, LSTM ve LSTM AutoEncoder'a kadar çeşitli ML ve DL sınıflandırıcıları kullanılmaktadır. Çeşitli modellerin dahil edilmesi, farklı veri kümeleri genelinde performansın daha kapsamlı bir şekilde değerlendirilmesine olanak tanır ve her modelin değişen bağlamlardaki etkinliğine ışık tutar.

Sonuçlar açısından, çalışmamız ISOT, FakeNewsNet, Dataset1 ve Dataset2 gibi veri kümelerinde ML sınıflandırıcıları için değişen ACC düzeylerini göstermektedir. Özellikle XGBoost modelinin tüm veri kümelerinde sürekli olarak yüksek performans göstermesi ve dikkat çekici ACC seviyelerine ulaşması dikkat çekicidir. Doğrusal SVM, performansta kayda değer değişkenlik sergiliyor ve RF sınıflandırıcı, FakeNewsNet veri kümesinde kayda değer ACC ile sağlam bir performans sunuyor. Bu, bu kadar çeşitli sınıflandırıcıları veya veri kümelerini keşfetmemiş olabilecek önceki çalışmalarla çelişiyor ve çalışmamızın alana benzersiz katkılarını ve içgörülerini sergiliyor.

DL alanında çalışmamız, farklı veri kümelerinde CNN, CNN-LSTM, CNN-GRU, LSTM ve LSTM AutoEncoder gibi modellerin performansındaki önemli değişkenliği ortaya koyuyor. Özellikle LSTM modeli, ISOT veri kümesinde mükemmele yakın ACC ile öne çıkıyor. İlginçtir ki, LSTM AutoEncoder modeli sürekli olarak muadillerine göre daha düşük performans sergiliyor ve bu da uygun DL modellerinin seçiminde veri kümelerinin spesifik özelliklerinin dikkate alınmasının önemini vurguluyor.

Çalışmamız FND hakkında giderek artan bilgi birikimine katkıda bulunuyor. Çeşitli veri kümelerini entegre etmesi ve çok çeşitli sınıflandırıcıları ve modelleri keşfetmesiyle kendisini farklı kılar. Sonuçlar FND'nin incelikli doğasını ortaya koyuyor ve dijital ortamda yanlış bilginin yol açtığı gelişen zorlukları ele almak için özel ve bağlama özgü bir yaklaşıma duyulan ihtiyacı vurguluyor.

4. Sonuç ve Bulgular

Özetle, kapsamlı araştırmamız, çeşitli uygulamalardaki yanlış bilgileri tanımlamaya yönelik çeşitli ML ve DL modellerinin potansiyeline ilişkin temel bir araştırma sağlar. Her modelin performansının kapsamlı bir değerlendirmesini sağlamak için her biri farklı özelliklere sahip çeşitli veri kümelerinden yararlandık. RF, Doğrusal SVM ve DT gibi geleneksel makine öğrenimi algoritmaları, CNN, LSTM ve hibrit versiyonlar gibi gelişmiş derin öğrenme çerçeveleriyle birlikte değerlendirildi. Analizimiz, bir modelin etkinliğini belirlemede veri

kümesinin ve sınıflandırma görevinin kritik rolünün altını çizdi. Bazı modeller tüm veri kümelerinde sürekli olarak yüksek ACC elde ederken, diğerleri belirli senaryolarda olağanüstü performans gösterdi. Bu, FND'nin karmaşık sorununu çözmeye yönelik evrensel bir çözümün bulunmadığını vurgulamaktadır. Bunun yerine sonuçlarımız, eldeki görevin özelliklerine özel olarak uyarlanmış modelleri birleştirerek en iyi sonuçların elde edilebileceği çok modlu bir stratejinin gerekliliğini vurgulamaktadır. Bu güçlü tekniğin desteğiyle, giderek dijitalleşen dünyada gerçeği daha iyi koruyabilir ve bilgi paylaşımının bütünlüğünü koruyabiliriz.

5. Gelecek İş

Hızla gelişen bu alandaki araştırma ortamı birçok heyecan verici fırsat sunmaktadır. İlk olarak, teknoloji ilerlemeye devam ettikçe sahte haber oluşturma ve yaymada kullanılan yöntemler muhtemelen giderek daha karmaşık hale gelecektir. Bu gelişen taktiklerin ilerisinde kalabilmek için modellerimizi sürekli olarak uyarlama ve geliştirme sürecini sürdürmemiz gerekiyor. Bir başka ilgi çekici araştırma yönü ise yanlış haberlerde sıklıkla gözlemlenen duygu analizi veya dil kalıpları gibi unsurların dahil edilmesidir. Bu eklemeler modellerimizin tahmine dayalı ACC'sini geliştirebilir. Çeşitli modellerin avantajlarını birleştiren topluluk yaklaşımlarının oluşturulması, yanlış bilgiye karşı daha güçlü bir bariyer oluşturmak için uygulanabilir bir plan sunmaktadır.

Bilim Kodu	: 902.1.014
Anahtar Kelimeler	: Sahte haber tespiti, makine öğrenmesi, derin öğrenme
Sayfa Adedi	: 219
Danışman	: Dr. Öğr. Üyesi Tuba ÇAĞLIKANTAR

ACKNOWLEDGEMENTS

Advisor who guided me with his valuable help and contributions throughout my studies

Dear Assist. Prof. Dr. Tuba AĞLIKANTAR, during my master's degree.

My mother, daughters, who supported me for never leaving me alone during the process,thank you for every thing.

I would also like to thank my friends.



CONTENTS

	Page
ABSTRACT.....	iv
ÖZET	v
ACKNOWLEDGEMENTS.....	xx
CONTENTS.....	xxi
LIST OF TABLES.....	xxiv
LIST OF FIGURES	xxv
LIST OF ABBREVIATIONS.....	xxxix
1. INTRODUCTION.....	1
2. CONCEPTS AND LITERATURE REVIEW	9
2.1. Introduction.....	9
2.2. Fake News.....	9
2.2.1. Fake News Definition	9
2.2.2. Fake news types	10
2.2.3. Fake news approaches	11
2.2.4. Platforms.....	14
2.3. Natural Language Processing	16
2.3.1. NLP Definition	16
2.3.2. NLP Techniques and methods	17
2.4. Machine Learning	19
2.4.1. Machine learning techniques	20
2.4.2. Machine learning algorithms	21
2.5. Deep Learning.....	26
2.5.1. Definition.....	26
2.5.2. Convolutional neural networks	26

	Page
2.5.3. Recurrent neural networks	29
2.6. Cross Validation	32
2.7. Related Work	33
2.8. Conclusion	36
3. METHODOLOGY	37
3.1. Introduction.....	37
3.2. Proposed Approach	37
3.2.1. Datasets description	38
3.2.2. Preprocessing	42
3.2.3. Machine learning based classifiers	43
3.2.4. Deep learning based classifiers	52
3.2.5. Evaluation measures	60
3.3. Conclusion	62
4. EXPERIMENTAL RESULTS.....	63
4.1. Introduction.....	63
4.2. The Training Parameters.....	63
4.3. Results of Binary Classification Using the ISOT Dataset	64
4.3.1. Machine learning classifiers results	64
4.3.2. Deep learning classifiers results	84
4.4. Binary Classification Using the FakeNewsNet Dataset.....	98
4.4.1. Machine learning classifiers results	98
4.4.2. Deep learning classifiers results	115
4.5. Results Using Dataset1	130
4.5.1. Machine learning classifiers results	130
4.5.2. Deep learning classifiers results	151
4.6. Results Using Dataset2	165

	Page
4.6.1. Machine learning classifier results	165
4.6.2. Deep learning classifiers results	185
4.7. Comparing Results	200
4.8. Discussion	203
5. CONCLUSION AND FUTURE WORK.....	205
5.1. Conclusion	205
5.2. Future Work	205
REFERENCES	207
CURRICULUM VITAE.....	219

LIST OF TABLES

Table	Page
Table 4.1. Results from ML classifiers on the ISOT Dataset	83
Table 4.2. Results from DL classifiers on the ISOT dataset.....	98
Table 4.3. Results from ML classifiers on the FakeNewsNet Dataset	115
Table 4.4. Results from DL classifiers on the FakeNewsNet Dataset	130
Table 4.5. Results of ML classifiers on Dataset 1	150
Table 4.6. Results of DL classifiers on Dataset 1	164
Table 4.7. Results of ML classifiers on Dataset 2	185
Table 4.8. Results of DL classifiers on Dataset 1.	200
Table 4.9. ACC scores of ML classifiers across various datasets.	201
Table 4.10. ACC scores of DL classifiers across various datasets.	203

LIST OF FIGURES

Figure	Page
Figure 2.1. Categories of ML algorithms	20
Figure 2.2. The structure of DTs.....	22
Figure 2.3. The algorithm for DTs.....	22
Figure 2.4. Classificatore NB.....	23
Figure 2.5. Decision boundaries and SVM margin.....	24
Figure 2.6. Basic layers of ANN.....	26
Figure 2.7. Design of CNN	27
Figure 2.8. Dropout Mechanism	29
Figure 2.9. Architecture of LSTM.	31
Figure 2.10. Architecture of GAN	32
Figure 2.11. K-fold cross validation	33
Figure 3.1. FND proposed framework.	38
Figure 4.1. Evaluation Matrix for RF classifier	65
Figure 4.2. Performance Evaluation of RF Classifier	66
Figure 4.3. Evaluation Matrix for GB.....	67
Figure 4.4. Performance Evaluation of GB Classifier	68
Figure 4.5. Evaluation Matrix for LR	69
Figure 4.6. Performance Evaluation of LR Classifier.....	70
Figure 4.7. Evaluation Matrix for XGBoost	72
Figure 4.8. Performance Evaluation of XGBoost Classifier.....	73
Figure 4.9. Evaluation Matrix for DT	74
Figure 4.10. Performance Evaluation of DT Classifier	75
Figure 4.11. Evaluation Matrix for MNB	76
Figure 4.12. Performance Evaluation of MNB Classifier.....	77

Figure	Page
Figure 4.13. Evaluation Matrix for SVM-RBF	79
Figure 4.14. Performance Evaluation of SVM-RBF Classifier.....	80
Figure 4.15. Evaluation Matrix for Linear SVM	81
Figure 4.16. Performance Evaluation of Linear SVM Classifier.....	82
Figure 4.17. Accuracy and loss metrics for training and validation of CNN	85
Figure 4.18. Evaluation Matrix for CNN	86
Figure 4.19. Classification Report of CNN	87
Figure 4.20. Accuracy and loss metrics for training and validation CNN-LSTM.....	88
Figure 4.21. Evaluation Matrix for CNN-LSTM.....	89
Figure 4.22. Performance Evaluation of CNN-LSTM Classifier	90
Figure 4.23. Accuracy and loss metrics for training and validation CNN-GRU.....	91
Figure 4.24. Evaluation Matrix for CNN-GRU	92
Figure 4.25. Performance Evaluation of CNN-GRU Classifier	93
Figure 4.26. Accuracy and loss metrics for training and validation LSTM	94
Figure 4.27. Evaluation Matrix for LSTM	95
Figure 4.28. Accuracy and loss metrics for training and validation LSTM-Autoencoder ...	96
Figure 4.29. Evaluation Matrix for LSTM-AutoEncoder.....	97
Figure 4.30. Evaluation Matrix for RF classifier.....	99
Figure 4.31. Performance Evaluation of RF	100
Figure 4.32. Evaluation Matrix for GB.....	101
Figure 4.33. Performance Evaluation of GB	102
Figure 4.34. Evaluation Matrix for LR	103
Figure 4.35. Performance Evaluation of LR.....	104
Figure 4.36. Evaluation Matrix for XGBoost	105
Figure 4.37. Performance Evaluation of XGBoost.....	106
Figure 4.38. Evaluation Matrix for DT	107

Figure	Page
Figure 4.39. Performance Evaluation of DT.....	108
Figure 4.40. Evaluation Matrix for MNB	109
Figure 4.41. Performance Evaluation of MNB	110
Figure 4.42. Evaluation Matrix for SVM-RBF	111
Figure 4.43. Performance Evaluation of SVM-RBF	112
Figure 4.44. Evaluation Matrix for Linear SVM.....	113
Figure 4.45. Performance Evaluation of Linear SVM	114
Figure 4.46. Accuracy and loss metrics for training and validation of LSTM.....	116
Figure 4.47. Evaluation Matrix for LSTM	117
Figure 4.48. Performance Evaluation of LSTM.....	118
Figure 4.49. Accuracy and loss metrics for training and validation of LSTM.....	119
Figure 4.50. Evaluation Matrix for CNN-LSTM	120
Figure 4.51. Performance Evaluation of CNN-LSTM	121
Figure 4.52. Accuracy and loss metrics for training and validation of CNN.....	122
Figure 4.53. Evaluation Matrix for CNN	123
Figure 4.54. Performance Evaluation of CNN	124
Figure 4.55. Accuracy and loss metrics for training and validation of LSTM-AutoEncoder	125
Figure 4.56. Evaluation Matrix for LSTM-AutoEncoder	126
Figure 4.57. Performance Evaluation of LSTM-AutoEncoder	127
Figure 4.58. Accuracy and loss metrics for training and validation of CNN.....	128
Figure 4.59. Evaluation Matrix for CNN-GRU.....	128
Figure 4.60. Performance Evaluation of CNN-GRU	129
Figure 4.61. Evaluation Matrix for RF.....	132
Figure 4.62. Performance Evaluation of RF.....	133
Figure 4.63. Evaluation Matrix for GB	134

Figure	Page
Figure 4.64. Performance Evaluation of GB	135
Figure 4.65. Evaluation Matrix for LR.....	137
Figure 4.66. Performance Evaluation of LR	138
Figure 4.67. Evaluation Matrix for XGBoost.....	140
Figure 4.68. Performance Evaluation of XGBoost	140
Figure 4.69. Evaluation Matrix for DT	141
Figure 4.70. Performance Evaluation of DT	142
Figure 4.71. Evaluation Matrix for MNB.....	143
Figure 4.72. Classification report of MNB.....	144
Figure 4.73. Evaluation Matrix for SVM with RBF	146
Figure 4.74. Performance Evaluation of SVM with RBF	146
Figure 4.75. Evaluation Matrix for Linear SVM.....	148
Figure 4.76. Performance Evaluation of Linear SVM	149
Figure 4.77. Accuracy and loss metrics for training and validation of LSTM.....	151
Figure 4.78. Evaluation Matrix for LSTM	152
Figure 4.79. Performance Evaluation of LSTM.....	153
Figure 4.80. Accuracy and loss metrics for training and validation of CNN-LSTM...	154
Figure 4.81. Evaluation Matrix for CNN-LSTM	154
Figure 4.82. Performance Evaluation of CNN-LSTM	155
Figure 4.83. Accuracy and loss metrics for training and validation of LSTM.....	156
Figure 4.84. Evaluation Matrix for CNN	157
Figure 4.85. Performance Evaluation of CNN	158
Figure 4.86. Accuracy and loss metrics for training and validation of LSTM-AutoEncoder	159
Figure 4.87. Evaluation Matrix for LSTM-AutoEncoder	160
Figure 4.88. CR of LSTM-AutoEncoder.....	161
Figure 4.89. Accuracy and loss metrics for training and validation of LSTM.....	162

Figure	Page
Figure 4.90. Evaluation Matrix for CNN-GRU.....	162
Figure 4.91. Performance Evaluation of CNN-GRU.....	163
Figure 4.92. Evaluation Matrix for RF	166
Figure 4.93. Performance Evaluation of RF	167
Figure 4.94. Evaluation Matrix for GB.....	168
Figure 4.95. Performance Evaluation of GB	169
Figure 4.96. Evaluation Matrix for LR	171
Figure 4.97. Performance Evaluation of LR.....	172
Figure 4.98. Evaluation Matrix for XGBoost	173
Figure 4.99. Performance Evaluation of XGBoost.....	174
Figure 4.100. Evaluation Matrix for DT	175
Figure 4.101. Performance Evaluation of DT.....	176
Figure 4.102. Evaluation Matrix for MNB	177
Figure 4.103. Performance Evaluation of MNB	178
Figure 4.104. Evaluation Matrix for SVM-RBF	180
Figure 4.105. Performance Evaluation of SVM-RBF	181
Figure 4.106. Evaluation Matrix for Linear SVM	182
Figure 4.107. Performance Evaluation of Linear SVM.....	183
Figure 4.108. Accuracy and loss metrics for training and validation of LSTM	186
Figure 4.109. Evaluation Matrix for LSTM.....	187
Figure 4.110. Performance Evaluation of LSTM	188
Figure 4.111. Accuracy and loss metrics for training and validation of CNN-LSTM ...	189
Figure 4.112. Evaluation Matrix for CNN-LSTM.....	190
Figure 4.113. Performance Evaluation of CNN-LSTM.....	191
Figure 4.114. Accuracy and loss metrics for training and validation of CNN	192
Figure 4.115. Evaluation Matrix for CNN.....	192

Figure	Page
Figure 4.116. Performance Evaluation of CNN.....	193
Figure 4.117. Accuracy and loss metrics for training and validation of LSTM Autoencoder	194
Figure 4.118. Evaluation Matrix for LSTM-AutoEncoder.....	195
Figure 4.119. Performance Evaluation of LSTM-AutoEncoder.....	196
Figure 4.120. Accuracy and loss metrics for training and validation of CNN-GRU	197
Figure 4.121. Evaluation Matrix for CNN-GRU	198
Figure 4.122. Performance Evaluation of CNN-GRU.....	199



LIST OF ABBREVIATIONS

The symbols and abbreviations used in this study are presented below, along with their explanations.

Abbreviations	Explanations
ACC	Accuracy
AI	Artificial Intelligence
ANN	Artificial Neural Network
CM	Confusion Matrix
CNN	Convolutional Neural Network
CR	Classification Report
DL	Deep Learning
DT	Decision Tree
F1	F1-Score
FND	Fake News Detection
GB	Gradient Boosting
GRU	Gated Recurrent Unit
LR	Logistic Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
NER	Named Entity Recognition
NLP	Natural Language Processing
PREC	Precision
REC	Recall
RF	Random Forest
RNN	Recurrent Neural Network
SM	Social Media
SVM	Support Vector Machine

1. INTRODUCTION

Context of the Thesis

In our dynamic and rapidly digitizing world, information access parameters have been remarkably expanded, reshaping our global society into an intricately interconnected digital village. The pervasive influence of digital technology has empowered individuals to effortlessly tap into an extensive reservoir of information from any corner of the globe with just a few clicks. As of January 2023, an astonishing 59.4% of the global population, approximately 4.76 billion individuals, actively participated in social media (SM) platforms, as elucidated in a comprehensive digital global report [1]. This statistic represents a substantial increase, with 137 million new users joining SM in the preceding year alone—a growth rate exceeding 3% annually, equivalent to more than four new users every second [1]. Recent studies underscore that over 90% of internet users engage with SM monthly, underscoring its pervasive influence [2]. The ascendancy of SM has not merely streamlined the dissemination of information; it has fundamentally transformed the global exchange of ideas and news. This paradigm shift has gradually diminished the reliance on traditional print media, which historically demanded rigorous research, comprehensive fact-checking, and detailed reporting to ensure reliability. While digital advancement brings myriad benefits, it also introduces significant challenges. The ease of accessing vast amounts of information, whether through computers or smartphones, presents a double-edged sword. The simplicity of circulating opinions, regardless of their factual accuracy or evidence, poses inherent risks. In extreme cases, this accessibility can be exploited for propaganda, manipulating public discourse to serve specific agendas. Among the most pressing concerns within the digital realm is the proliferation of fake news [3]. Effectively addressing this issue necessitates clearly defining "fake news" as content intentionally deceiving readers into believing falsehoods [4]. Propagated through prominent SM platforms, emails, websites, and even radio services, fake news is a pernicious form of propaganda, wielding influence over public opinion [5]. Characterized by its inauthenticity and the intent to mislead, fake news often comprises false information that, despite being easily verifiable, is crafted to deceive readers. The repercussions of rapidly disseminated online misinformation can be profound, encompassing the erosion of trust in the news ecosystem, the tarnishing of reputations, and the cultivation of public fear and skepticism. If left unchecked, these impacts can potentially contribute to societal instability [4].

While social media (SM) undeniably offers the advantages of cost-effective and swift news dissemination, the lax regulation of the internet amplifies the proliferation of misinformation. The unchecked circulation of false information can disrupt the equilibrium of authenticity within the ecosystem, coercing individuals into embracing biased or inaccurate perspectives. In this digitally interconnected era, heightened awareness and vigilance against the insidious spread of fake news become imperative. Fake news detection (FND) encompasses three intricate subcategories: fabrication, hoax, and satire. Fabrication entails the craft of sensational headlines and the intentional omission of information from unreliable sources, creating a distorted narrative. Hoaxes employ more sophisticated deception techniques, leveraging multiple channels for propagation to enhance credibility among the public. Satire, on the other hand, introduces fictitious information under the guise of humour, a nuanced form of misinformation that, if misunderstood, can be misconstrued as factual, further complicating the landscape of discerning truth from deception.

Significance of the Thesis

This research delves into the profound potential for elevating our capabilities in identifying and mitigating the detrimental impacts of false news and misinformation. The consequences of disseminating false news are far-reaching and severe, ranging from erasing public trust to instigating violence. It becomes imperative to develop precise methodologies that can effectively discern and spotlight false news to uphold the integrity of information and ensure public access to trustworthy and accurate sources.

This study aspires to pinpoint the most effective techniques for false news detection. By meticulously analysing each model's performance in identifying false news and leveraging the widely accessible ISOT dataset, the research endeavours to contribute to creating more resilient and efficient tools and methodologies for false news detection. These advancements play a pivotal role in mitigating their influence regarding inaccurate information and disinformation, thereby elevating the quality and reliability of available information for the public.

Objectives of the Thesis

The development and assessment of models and methodologies for FND is the main focus of this thesis. The main goal is to develop accurate and effective algorithms to identify and classify misleading or incorrect content on social media and news stories, among other digital media. This

involves thoroughly examining textual content and relevant features to discern genuine information from fabricated claims, addressing biases, propaganda, and other manipulation techniques.

This study centrally revolves around evaluating and comparing the efficacy of various ML and DL models in detecting false news using an openly available dataset. Through the implementation and critical analysis of these models, the research aims to gauge their accuracy in identifying false news and understand each approach's inherent strengths and weaknesses. The evaluation considers factors that offer insights into the models' ability to distinguish between factual and non-factual information.

The main objective of this research is to develop false news detection methodologies significantly. The performance analysis of the suggested models is expected to yield insights that will guide the creation of more advanced and potent FND tools and tactics. The ultimate goal is to increase the automation of systems' capacity to detect and prevent the propagation of misleading information, hence promoting the public's access to reliable and correct news.

Challenges of the Thesis

The field of FND research is fraught with many challenges, introducing intricate layers of complexity to the overarching issue. Understanding and addressing these challenges are crucial for formulating effective solutions.

A noteworthy challenge in FND is fake news's dynamic and constantly evolving nature. False information takes various forms, encompassing doctored news articles, manipulated images, misleading headlines, and pervasive rumours. Moreover, those disseminating fake news adapt their strategies to keep pace with evolving technological advancements and new media platforms, making developing a universally applicable and comprehensive approach a formidable task. Researchers continually refine and augment their models and methodologies to stay abreast of the dynamic trends in fake news.

Another impediment in FND research is the absence of standardized datasets and consistent evaluation metrics. The sensitive and contentious nature of false news complicates the compilation of a comprehensive and representative dataset, hindering effective comparisons and benchmarking

of various FND models and approaches. The lack of uniform evaluation metrics further complicates the precise assessment of these models' effectiveness. Progress in the field of FND requires the establishment of robust evaluation frameworks and benchmark datasets, facilitating fair and accurate comparisons across different methodologies.

Ethical and legal considerations present substantial challenges in FND research as well. Balancing the identification and counteraction of fake news while respecting fundamental rights such as free speech and privacy and preventing censorship is complex. FND systems must be intricately designed to uphold user rights and maintain transparency in their detection processes. Achieving this balance demands a thoughtful approach to ethical issues and stringent compliance with legal standards, ensuring that the development and application of FND systems align with societal values and legal norms.

Motivation of the Thesis

This thesis is motivated by growing worries about the spread of misleading information and its possible negative effects on people and society. Spreading incorrect information can seriously damage people's faith in institutions, upend societal cohesion, and, in extreme situations, spark violence and unrest. It is critical to address this issue, which calls for developing practical tools and tactics to thwart misleading information and give the public access to accurate and dependable information.

The recent evolution of artificial intelligence (AI) presents promising avenues for tackling the false news dilemma. AI technologies can analyze extensive data from diverse sources, identifying behavioural patterns indicative of the dissemination of false information. These technologies play a crucial role in detecting and flagging misleading or inaccurate content, thereby mitigating the negative impact of false news. Natural Language Processing (NLP), a pivotal component within AI, enables the processing and interpretation of human language in a manner understandable to machines for analysis. Applying ML models and DL techniques to false news detection provides a pathway for developing effective methods to identify and counteract false news.

From massive datasets like the specially created ISOT dataset for false news detection, ML and DL models are excellent at discovering trends and extracting relevant information. By training these models on such datasets, one can more easily identify distinctive characteristics and signals

that point to false news, improving the accuracy of detection techniques. Moreover, the identification process can be made more efficient and scalable by automating and streamlining it using DL in false news detection.

The fundamental motivation of this research is to harness the potential of AI, NLP, ML, and DL in addressing the critical challenge of false news. This thesis aspires to contribute novel algorithms and models to false news detection, augmenting the capability to identify and counteract false information effectively and expeditiously. The use of AI technology in this endeavour has the potential to transform the detection of false news, offering crucial instruments to maintain the accuracy of information distribution and protect people and communities from the damaging effects of incorrect information.

This thesis is primarily motivated by the urgent need to combat the growing threat of false news, which puts people and society at serious risk. This project aims to advance the field of false news detection by utilizing AI, NLP, ML, and DL techniques. It focuses on developing effective and automated strategies for detecting and refuting misleading information. The ultimate goal is to reduce the negative consequences of false news by cultivating a more resilient and knowledgeable society that can make decisions based on trustworthy information.

Contribution of the Thesis

This thesis makes substantial strides across several pivotal domains, encompassing the enhancement of news quality, the reduction of false information dissemination, and significant contributions to NLP and ML.

The study of false news identification is primarily essential to improving the calibre and dependability of news and information that the general public can access. In a time when news is consumed primarily online, it is critical to guarantee the veracity and correctness of shared information. This is essential to promote informed decision-making and active engagement in democratic processes. The novel techniques and models for identifying false news that this study has produced help to improve the caliber of news reporting and foster an informed and trustworthy public sphere.

Second, addressing and preventing dissemination of inaccurate and misleading information is critical. False information can have severe consequences for people, companies, and society. It can sway public opinion, warp perceptions, and exacerbate neighbourhood tensions. Using the publicly available ISOT dataset, this thesis presents three cutting-edge deep learning algorithms designed to detect false news. These contributions are essential in preventing the dissemination of misleading information, which lessens the negative consequences of its dissemination and protects the integrity of public debate and the welfare of society.

Finally, this work advances the disciplines of NLP and ML by developing new approaches to text analysis and categorization in identifying noteworthy false news. To combat false news detection effectively, new approaches are needed to distinguish between reliable and deceptive sources of information. The DL algorithms presented in this thesis enhance the methods used in NLP and ML by offering new insights into text analysis and classification. This work opens the door for future research and the development of more advanced algorithms and models in these domains, in addition to adding to the body of knowledge already in existence.

Thesis Report Organization

The structure of this thesis unfolds as follows:

1. Chapter 2 offers a foundational overview, encompassing fundamental concepts and reviewing pertinent literature.
2. In Chapter 3, the thesis delves into the proposed methodology for detecting fake news. It provides a detailed description of the dataset used, discusses the implementation of the suggested deep learning (DL) models, and concludes by elucidating the evaluation metrics employed to gauge the performance of these models.
3. Chapter 4 is dedicated to illustrating and analyzing the outcomes of various classifiers, offering a meticulous examination of their results.
4. Lastly, Chapter 5 concisely summarises the research conducted in this thesis and delineates potential paths for future exploration in this field.

Conclusion

This chapter provides a concise synopsis of the thesis report, emphasizing the research's main difficulties, driving forces, goals, and overall importance. The next chapter will provide a detailed description of the various fields that we use to accomplish our study goals. It will also encompass a comprehensive examination of existing research that employs a variety of ML and DL models in combination with diverse pre-processing datasets and methods to detect false news proficiently.





2. CONCEPTS AND LITERATURE REVIEW

2.1. Introduction

This chapter explores the fundamental ideas relevant to our FND study. The concept of fake news and its different applications, kinds, and contexts are first explored. After that, we look at the NLP fundamentals. The chapter then describes the underlying ideas and methods related to ML and DL. The final section of the chapter offers a thorough analysis of connected works.

2.2. Fake News

2.2.1. Fake News Definition

This process can be envisioned as a two-phase cycle involving news creation and consumption. On the creation side, news publishers operate with both immediate and long-term objectives. In the short term, their focus is often on maximizing profits, which is achieved by broadening their consumer reach. In the long term, the goal shifts towards establishing and upholding a reputation for delivering trustworthy news. On the consumption side, the consumer's needs are twofold. They seek accurate and straightforward information and news that aligns with their pre-existing beliefs and social expectations. Theories like the social identity theory and confirmation bias, which contend that social validation and avowal are integral to a person's identity and self-esteem, provide context for this latter demand [6]. Fake news tends to thrive when consumers prioritize psychological needs over the desire for balanced and accurate information. This happens as publishers focus on short-term gains while consumers are driven by their psychological desires [7].

Fake news is characterized as intentionally crafted content with malicious intent to mislead readers for various commercial, political, and social purposes [8]. This deceptive content can manifest in various forms, such as fabricated articles, misleading headlines, manipulated photos or videos, and the impersonation of credible news sources [9], [10]. Disseminating false information can adversely affect individuals and communities and may even disrupt political processes and societal stability [12].

The surge in internet usage and the growing number of social media (SM) users in recent years have increased attention to this issue [13]. To effectively counter fake news, critical examination of information sources and diligent fact-checking before dissemination are essential [14]. Moreover, fostering media literacy and educating individuals on identifying false information are crucial steps towards navigating the complex information landscape more effectively [15].

A prominent example of the rapid spread of misinformation is the 2020 coronavirus pandemic, a significant global health crisis comparable to the 1918 Spanish flu in its scope and impact [16]. As documented in Nature [17], the pandemic has become a breeding ground for diverse forms of misinformation. This spans from dangerous health recommendations, such as suggesting the use of disinfectants as intravenous treatment, to bizarre narratives like Russian President Vladimir Putin releasing lions to enforce lockdown orders.

2.2.2. Fake news types

There are three primary categories of fake news identifiable [13]:

- **Fabrication:** This type of fake news involves creating content deliberately to deceive the audience [13]. Fabricated news stories often lack factual accuracy, as they involve altering or omitting facts to construct a misleading narrative [18]. These stories typically stem from one or a few individuals interested in disseminating false information. To engage readers, such articles frequently use eye-catching headlines and sensational content [19].

A comprehensive strategy is required to battle fake news properly. Fact-checking, which involves confirming the accuracy and reliability of the material in news pieces, is an essential step. Organizations that verify facts are crucial in examining statements, verifying information from multiple sources, and providing the public with accurate information. Fact-checkers can identify errors, contradictions, and skewed information in fake news by carefully examining the material [20].

Furthermore, technological developments, such as NLP and ML algorithms, have significantly contributed to detecting fabricated news [21]. These automated systems can analyse the text in news articles, identify language patterns indicative of manipulation or misinformation, and mark potential falsehoods [22]. By being trained on extensive

datasets of verified news articles, these systems learn to recognize the traits and patterns typical of fabricated content, enhancing their ability to detect dubious material more accurately [22].

- Hoax: Hoaxes are a more complex variant of fake news, utilizing elaborate deception techniques to mislead the public [13]. Contrasting with fabrication, hoaxes typically involve the distribution of the same falsified news story across multiple outlets, making detection and refutation more difficult. The widespread propagation of hoaxes through various websites, social media (SM) platforms, and other online media heightens their potential acceptance by the broader audience [23].

A thorough approach is needed to detect and address hoaxes, including cross-verifying sources and consulting several trustworthy outlets. Assessing the trustworthiness and accuracy of information is crucial in differentiating between legitimate news and hoaxes. Fact-checking organizations are instrumental in rigorously examining the claims made in these reports, confirming the information with reliable sources, and uncovering the falsehoods.

- Satire is another notable category of fake news [9]. It involves intentionally crafting news stories with a humorous or ironic twist, not intending to mislead but rather to entertain or offer commentary on real-life events [24]. However, those unfamiliar with its source or context can misinterpret satire as factual, leading to its unintentional spread as legitimate news.

2.2.3. Fake news approaches

This study presents the results of research conducted by various scholars on identifying false news utilizing diverse methods:

- Visual-based: In the era of social media (SM) dominance, the issue of visual-based fake news has risen to prominence, given its capacity to swiftly reach a broad audience and potentially influence public perception, decision-making, and social and political dynamics [8]. The editing or manipulating of photos and videos is a crucial component of visual-based false news [25]. This can involve using advanced editing techniques to alter the visual media's content and produce false depictions of places, people, or events. Altered images or videos can distort the truth, support false narratives, or create fictitious situations. Using visual media, perpetrators can mislead viewers and influence their

emotions, viewpoints, and beliefs. Another strategy in visual-based fake news is contextual manipulation [5].

When visual content is presented without sufficient context, false news producers can influence how viewers perceive and understand their content. Content can be significantly altered in meaning when taken out of its original context, resulting in misconceptions or misinterpretations. The original message intended by a visual artist may be distorted through cropping, altering, or pairing with misleading narratives. With digital editing techniques and technologies developing rapidly, it is becoming more difficult to distinguish between original and modified visual content, making it harder to detect doctored graphics. So, before accepting and sharing visual content, people must develop critical thinking skills and fact-check to ensure that it is genuine and accurate.

- **User-based:** This tactic entails disseminating inaccurate or misleading information to particular people or groups according to their beliefs, interests, or demographics [26]. The objective is to influence and mould people's views and actions by producing content that conforms to their preexisting prejudices or preferences. Using false accounts or SM platform bots is a common strategy in user-based fake news [8]. These organizations imitate actual user behaviour, disseminating misleading information with a sense of legitimacy and reliability. Fake news based on user activity can infiltrate online communities and interact with people to effectively shape public opinion and direct conversations within particular groups. User-based false news uses data on users' interests, behaviours, and digital interactions to target content to specific persons by utilizing the personalized aspect of online platforms, where algorithms and user data help [90]. This makes it possible for those who fabricate fake news to produce and disseminate information that specifically targets the tastes and worldviews of their intended audience [91]. This customized strategy increases the likelihood of misinformation being accepted and spread among the target audience. User-generated fake news has serious consequences, exacerbating preexisting prejudices, creating echo chambers, and deepening societal division [92]. This results in the spread of false information, a reduction in trust in reliable information sources, and the maintenance of narratives that divide people. Furthermore, user-generated fake news can affect actual events by swaying public opinion, election results, and societal norms.
- **Network-based:** This strategy entails transmitting incorrect or misleading information through social networks and online communities, where connections are made based on personal or professional contacts [93]. This strategy uses social networks'

interconnectedness to spread bogus news more widely and powerfully. In network-based fake news, people or organizations intentionally or unintentionally spread misleading information within their networks, frequently by sharing messages, posts, or links that include misleading or fraudulent content [94]. Coordinated attempts by entities may occasionally aim to spread false information to influence public opinion or accomplish particular goals [69]. Using a network's innate biases and preconceptions is key to network-based false news. Fake news creators can create content that appeals to the members of a network by appealing to their shared values, beliefs, or interests. This allows false information that supports the group's preexisting narratives or ideas to spread quickly [93]. The dissemination of fraudulent accounts and bots on networks is further accelerated. Bots are automated SM accounts designed to mimic human behaviour and spread fraudulent information under the guise of widespread support or consensus. The fabrication of fictitious accounts seeming to be actual people lends legitimacy to the misleading information being spread [95]. The effects of network-based fake news are profound, exacerbating societal divisions and undermining confidence in trustworthy sources of information. False information transmitted widely over a network can strengthen prejudices and create echo chambers where factual information or opposing viewpoints are ignored [27]. The integrity of democratic systems, public debate, and decision-making procedures may all suffer from such dynamics.

- **Post-based:** This category includes the sharing of erroneous or misleading content via individual SM posts, such as tweets, videos, photo captions, and memes [28]. These posts are meant to catch attention, and to do so, they frequently use sensationalism, provocative content, or shock value. The main goal of post-based fake news is to increase interaction and spread via SM platforms [28]. These postings use attention-grabbing headlines, clickbait techniques, and visually appealing material to attract readers and encourage them to tell their networks about the content. Fake news with a post-based origin can have a significant impact. The public may be tricked and misinformed by the spread of false information, which could lead to misunderstandings, false beliefs, and confusion. This may harm public opinion, skewing people's beliefs and behaviour [95]. Fake news that is post-based can further deepen political and social divisions. These postings can potentially exacerbate ideological echo chambers and polarize society because they are consistent with preconceived notions and biases. They may provoke hatred enmity and widen rifts between various groups [95]. Identifying the difference

between authentic and fake news in the SM era poses a significant challenge [90]. The prevalence of clickbait and eye-catching headlines complicates users' ability to distinguish trustworthy from misleading or fabricated content. The rapid and often unchecked dissemination of information on SM platforms further complicates efforts to combat post-based fake news.

2.2.4. Platforms

- **Social media:** The most prevalent method for news dissemination on SM platforms involves users sharing the content. Popular social networking sites are commonly used for spreading information [29]. For example, the potential for broader reach is amplified when false information is paired with sponsored ads on platforms like Facebook, allowing for posts with specific character lengths. These posts can be further propagated as they are shared and re-tweeted among Twitter users, increasing their audience reach.
- **Standalone website:** Each website, possessing its unique URL, allows individuals to publish new stories. These URLs are often directly used for posting on SM platforms. The internet predominantly consists of blogs, media sites, and renowned news websites [31]. Authentic content is typically associated with well-established news websites with a significant SM presence. In contrast, blog sites mainly depend on user-generated and unregulated content and are prime venues for spreading false information. Media sites, characterized by content-rich offerings, enable users to create websites and produce content that aligns with specific client preferences and styles.
- **Radio service:** Maintaining radio services' credibility and dependability depends on ensuring the material they transmit is accurate. Radio services are significant local news sources. Publications like *Wireless Communications* and *Mobile Computing* underscore the importance of secure and verified information dissemination for an informed and safe community [128].
- **Emails:** While emails are often perceived as a reliable source of news, verifying the authenticity of news-related emails can be challenging [129]. The credibility of email as a news medium lies in its direct and personal nature, but this also makes it a potential conduit for disseminating unverified or false information.

News data types

In the news dissemination landscape, various formats and platforms are employed to provide information to consumers. Here are the four main methods through which news is typically consumed:

- 1- **Audio:** Employed as an independent platform by news outlets, audio includes traditional broadcasts and podcasts. It enables people to access news while involved in other activities, like driving. Utilizing spoken words and interviews, audio news effectively communicates stories, capturing the listener's attention through auditory engagement [130].
- 2- **Multimedia:** This format integrates diverse media elements such as music, images, videos, and graphics [131]. The amalgamation of these elements makes news content visually stimulating. Multimedia news leverages visual aids to narrate intricate stories, evoke emotions and thoroughly understand subjects. This approach lets news providers present information dynamically and engagingly, appealing to a broad audience [132].
- 3- **Hypertext Links and Embedded Material:** Hypertext links are crucial in news consumption, enabling users to explore additional information sources [133]. News articles provide backing evidence and additional context or guide readers to related content. Embedded material, increasingly familiar with the rise of SM, can include SM posts, SoundCloud audio, YouTube videos, etc., within news articles. These elements enrich news stories, offering readers direct engagement with diverse perspectives and multimedia from various sources [134].
- 4- **Text:** Written content remains a core medium for news communication. Text analysis focuses on understanding the meaning conveyed through language, considering aspects like discourse analysis, grammar, tone, and pragmatics [135]. Textual news articles offer comprehensive information, insights, and interpretations of events, with journalists aiming to present accurate, fair, and informative written content [136].

These varied types of news data cater to user preferences, offering multiple engagement modes with information. Whether they use text, audio, multimedia, hyperlinks and embedded files, or another medium, news providers want to present engaging, educational, and easily accessed content.

2.3. Natural Language Processing

2.3.1. NLP Definition

NLP explores the relationship between computer technology and human language in an exciting and rapidly evolving field [32]. NLP has grown significantly and is being used in more sectors due to advances in ML and the expanding availability of large amounts of textual data [33].

NLP is used for a wide range of applications and tasks. One crucial task is text categorization or classification, which is helpful in fields such as document sorting, sentiment analysis [35], topic modelling, spam detection [34], and sentiment analysis [35]. It involves categorizing or labelling text documents automatically according to their content. One of the most important applications of NLP is language translation, which aims to translate text mechanically between languages to facilitate successful communication [36].

Named Entity Recognition (NER), a key element of NLP, is concerned with locating and retrieving certain entities from text, including people, places, dates, and organizations. Information extraction, entity disambiguation, and semantic analysis are some activities where NER is beneficial [37].

NLP text summarizing attempts to provide concise summaries from long text documents, valid for information retrieval, news aggregation, and document summation [38]. NLP includes speech synthesis and recognition. These technologies allow for the conversion of spoken language into text and vice versa, opening up a world of possibilities for applications like interactive voice response systems, voice assistants, and accessibility aids [39].

NLP leverages diverse computational methodologies and algorithms. ML is pivotal in NLP [40], facilitating the learning of patterns and extraction of insights from text. DL, a branch of ML, is particularly notable for its capacity to learn complex linguistic patterns and hierarchical representations. DL models like RNN and transformers have achieved groundbreaking results in numerous NLP tasks.

Furthermore, NLP significantly depends on linguistic resources like lexicons and syntactic parsers for more in-depth language analysis and comprehension. In the closely related

discipline of computational linguistics, linguistic theories and computer methods are combined to create models and algorithms to process linguistic phenomena.

2.3.2. NLP Techniques and methods

Tokenization

Tokenization is a crucial process in NLP involving the division of text into smaller segments known as tokens. These tokens can represent entities ranging from single words to phrases and even entire sentences. The purpose of tokenization is to split the text into units that hold meaning, allowing for more detailed analysis or further processing [41]. In NLP, tokens are the most basic independent units of natural language, forming the foundational elements for various language-processing tasks [42]. NLP systems can better understand a text's structure, meaning, and context by breaking it down into tokens, which makes for improved analysis and interpretation [43], [44]. Tokenization uses various techniques and algorithms, each selected according to the particular requirements and characteristics of the textual material. The following are a few popular tokenization methods:

- Word Tokenization: involves breaking down a text into its words and considering each word as a separate token [137]. For instance, applying this technique to the phrase "Google is amazing" would result in four separate tokens: "Google," "is," and "amazing." This approach is widely used in various NLP applications, including but not limited to text categorization, sentiment analysis, and language modelling, making it a fundamental and widely applied method in the field.
- Phrase Tokenization: Treating phrases or multiword utterances as a single token is advantageous [138]. For instance, it could be more accurate to regard "New York City" as a single token instead of breaking it into three words. Phrase tokenization is crucial to preserve the context and meaning of these multi-word statements throughout the analytic process.
- Sentence Tokenization: This method divides the text into individual sentences, each regarded as an independent token [139]. Sentence tokenization is especially relevant in scenarios that require analysis at the sentence level or in handling text corpora that are structured sentence by sentence.

Tokenisation algorithms can vary in complexity and sophistication depending on the particular requirements of the NLP work at hand. More straightforward methods like regular expressions or rule-based algorithms are frequently adequate for basic tokenisation. These methods rely on criteria such as delimiters, whitespace, punctuation, or other linguistic markers to detect and segment tokens using predetermined rules and patterns [137].

In the realm of FND, tokenization plays a critical role. It involves breaking down text from news articles, SM posts, and other textual formats into smaller, meaningful units for further analysis and processing. Segmenting the text into tokens allows researchers and algorithms to understand better the text's structure, semantics, and context. This understanding is vital for recognizing patterns and characteristics typical of fake news [140].

Stemming and lemmatization

Lemmatization and stemming are pivotal processes in NLP, designed to simplify words to their most basic form, which helps enhance the efficiency of NLP models by reducing the complexity of the text [32].

The more complex of the two, lemmatization, entails dissecting a word's morphological structure to determine its lemmas or dictionary form. To accurately change the term, this technique considers several characteristics, including its part of speech and contextual usage. For example, "running" would become "run" when lemmatized. Lemmatization ensures that a word's many inflected forms are combined into one basic form, which helps to reduce repetition and improve textual data analysis [141].

On the other hand, stemming is a more straightforward, rule-based technique that involves removing words' prefixes and affixes to reveal their stem or root form. It removes common word ends using a set of heuristics, frequently without considering context or part of speech. For instance, by eliminating the "-ning" suffix, "running" would become "run" through stemming. Even though stemming is more computationally efficient and quicker, it might not always provide meaningful or valid word forms [142]. Although this method is typically less accurate than lemmatization, it may work well in some situations where simplicity and quickness are key considerations.

Part-of-speech tagging

Part-of-speech (POS) tagging is a process where each word in a sentence or phrase is assigned a specific POS tag [96]. This technique is instrumental in identifying the grammatical role of each word, shedding light on its syntactic function within the sentence.

POS tagging is a foundational tool in various NLP tasks, enhancing the depth and precision of language analysis. For instance, in syntactic parsing, which involves dissecting the structural arrangement of sentences, POS tags are essential for delineating word relationships and constructing syntactic parse trees. In NER, an important NLP task, POS tagging aids in identifying and categorising named entities like people, organizations, locations, and dates [97]. Furthermore, in text summarization, POS tags are utilized to pinpoint and extract key information, aiding in the creation of brief summaries.

Multiple methodologies and algorithms for POS tagging exist, each with unique strengths and limitations. Rule-based algorithms rely on meticulously crafted rules and patterns, assigning POS tags based on specific linguistic cues, including word suffixes, prefixes, contextual usage, and syntactic relationships. These rules are designed to interpret and categorize words based on their grammatical properties and surrounding linguistic context.

2.4. Machine Learning

ML is employed to enhance machines' capability in processing and understanding data more efficiently. There are instances where the information within data remains elusive despite thorough analysis, and it is in these scenarios that ML proves particularly valuable. The increasing availability of large datasets has escalated the demand for ML applications. Various organizations leverage ML to distil relevant information from their data troves. ML facilitates learning from data, enabling machines to develop their understanding autonomously without explicit programming. To address the challenges posed by vast datasets, numerous mathematicians and computer scientists have explored and implemented various methods and techniques [98]. These efforts focus on devising ways to enable machines to learn and adapt independently through data analysis and pattern recognition.

2.4.1. Machine learning techniques

AI incorporates a diverse array of ML techniques, each with its unique advantages and limitations. The selection of a particular ML approach is contingent upon the nature of the problem and the extent to which labelled data is available [143]. Figure 2.1 presents an overview of the different types of ML methodologies.

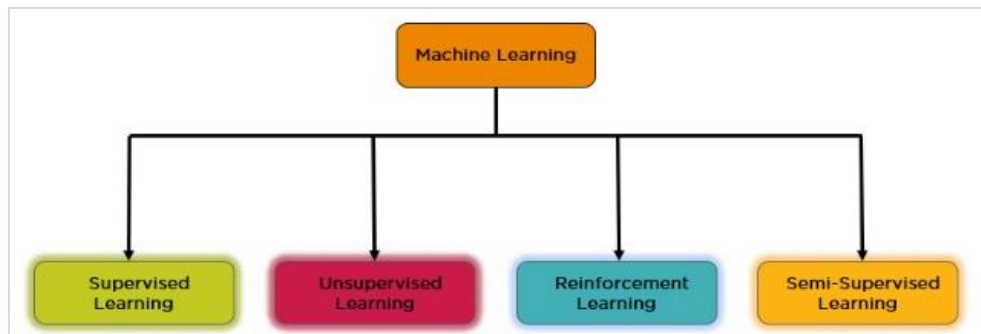


Figure 2.1. Categories of ML algorithms

Supervised learning

In supervised learning, an algorithm is trained using a dataset with labelled input data and corresponding desired outcomes. The algorithm utilizes this labelled dataset to identify patterns and relationships between inputs and outputs, subsequently applying this knowledge to predict outcomes for new, unseen data. This learning type is particularly prevalent in classification and regression tasks [99].

Unsupervised learning

Unsupervised learning entails training a computer with a dataset that lacks labelled output data. In this approach, the algorithm explores the data to discover inherent correlations and patterns, subsequently organizing the data based on these insights. Key applications of this learning type include clustering and dimensionality reduction [100].

Semi-Supervised learning

Semi-supervised learning is an amalgamation of supervised and unsupervised learning methods. In this approach, a computer is trained with a partially labelled dataset - some input

data come with known outcomes, while others do not. Initially, the algorithm learns from the labelled data and then extrapolates this learning to make predictions for the unlabeled data [101].

Reinforcement learning

In reinforcement learning, a computer learns through trial and error, receiving feedback as rewards or penalties based on its actions. The algorithm uses this feedback to improve its decision-making capabilities progressively. This type of learning is commonly used in environments where a system needs to interact dynamically with its surroundings to achieve specific goals, such as in robotics, gaming, and interactive systems [102].

2.4.2. Machine learning algorithms

Random forest

The Random Forest (RF) Algorithm is an ML method that integrates predictions from multiple trained classifiers. Primarily used for classification or regression tasks, it is particularly suited for decision tree (DT) classifiers. In this approach, a collection of DTs is randomly sampled from subsets of the training data. The final classification of a test item is determined by aggregating the votes from these multiple DTs [45].

Decision tree

DTs are widely used in data mining for predictive modelling (Figure 2.2). A DT is structured with nodes and leaves, where each node represents a decision point examining a specific data feature, leading to data division accordingly [46]. The leaves signify the predicted values for a data point. DTs operate on a recursive principle; their construction does not rely on assumptions like normal distribution. They also overlook collinearity or correlation among explanatory variables [47]. Each node in the tree corresponds to an attribute of the data, with branches representing potential outcomes or values of that attribute [48]. The tree makes predictions about the target variable based on the provided input data by tracing the path from the root to a leaf. Whereas decisions in regression trees are made to minimize the sum of squared residuals, those in classification trees are directed at each node by the feature that offers the most significant information gain or the most substantial reduction in entropy.

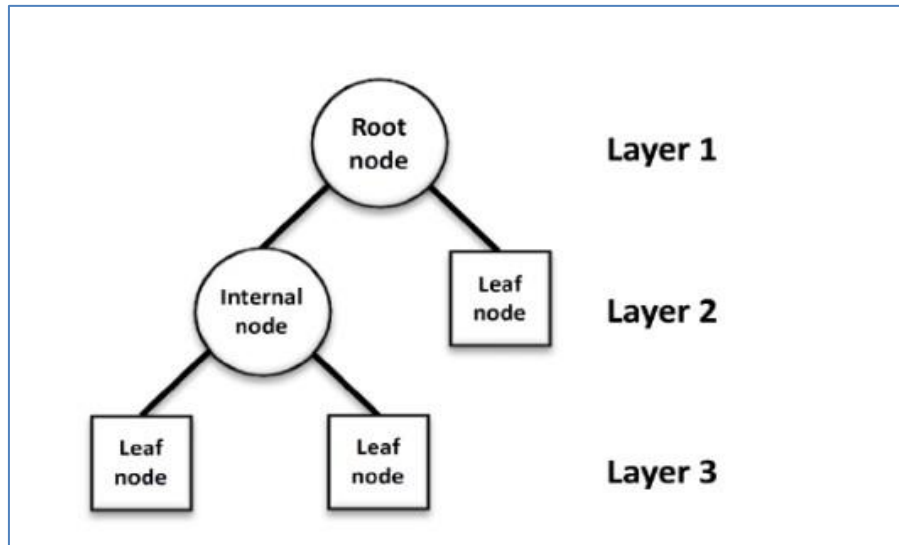


Figure 2.2. The structure of DTs [125].

The diagram of the DT algorithm is depicted in the subsequent Figure 2.3.

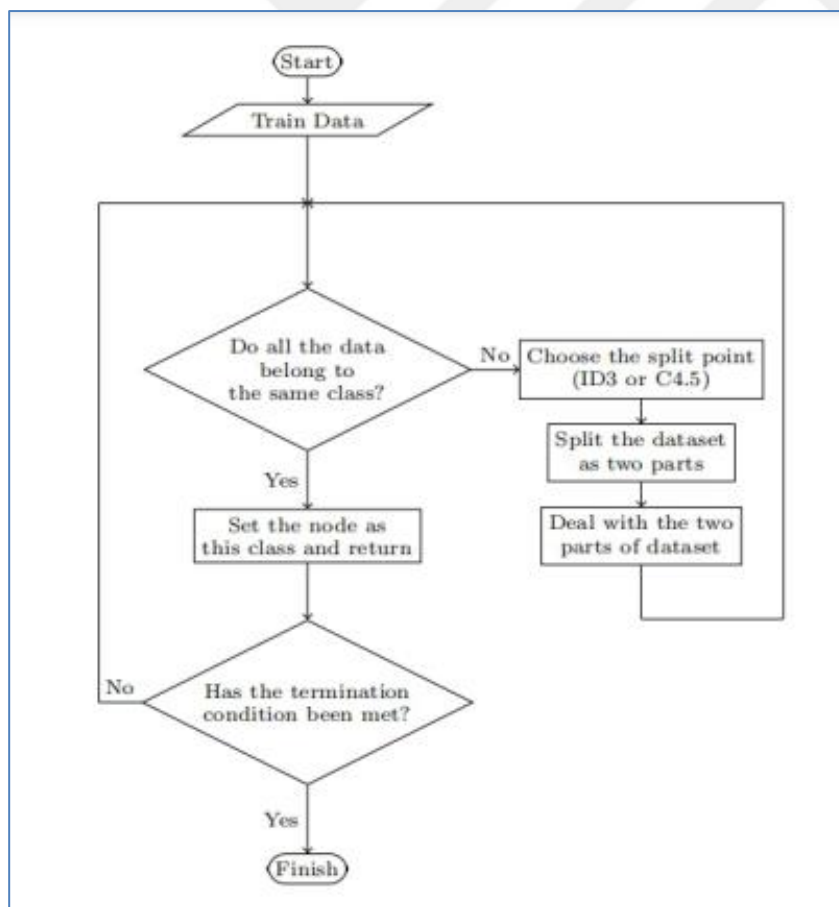


Figure 2.3. The algorithm for DTs [127].

Naive Bayes

The NB algorithm stands as one of the earliest and most recognized techniques in the realm of ML [49]. This algorithm operates on the naive assumption of independence among variables, simplifying the process for probability-based classification calculations. Although this assumption oversimplifies real-world scenarios, the NB algorithm has demonstrated effective performance when applied to actual databases, mainly when used in conjunction with an attribute selection method.

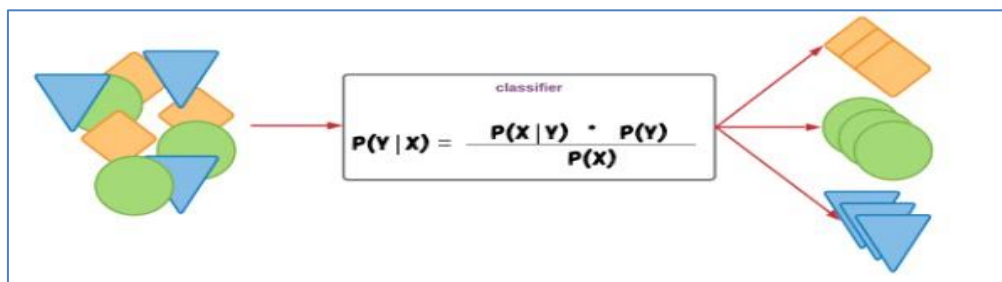


Figure 2.4. Classificatore NB [126].

Support Vector Machines

Support Vector Machines (SVMs) represent a supervised learning approach suitable for regression and classification [50]. The fundamental concept of SVMs involves employing the kernel method to transform the initial data into a higher-dimensional space, facilitating data segmentation through a hyperplane. In classification applications, the primary goal is to identify a hyperplane that effectively separates distinct classes in the dataset, maximising the margin—the distance between the hyperplane and the nearest data points [51].

In SVM algorithms, the Radial Basis Function (RBF) kernel stands out as a frequently employed kernel function [52]. It gauges the similarity between two points by assessing their distance in the original feature space. The RBF kernel is recognized for its smooth, bell-shaped curve that diminishes with distance from its centre. The shape and attributes of the kernel function can be adjusted by manipulating the gamma and C parameters of the RBF kernel.

The hyperplane functions as a linear decision boundary in linear SVMs [52], which divides the classes. In essence, linear support SVM identifies the line (or hyperplane) that most

effectively partitions the data points into discrete groups. The maximum margin between this line and the closest points in each class is maximized, referred to as the maximum-margin hyperplane. Although linear SVMs may accommodate non-linear boundaries through the kernel method, they are best suited for linearly separable data [53]. The following equation defines the decision boundary in an SVM:

$$w \cdot x + b = 0 \quad (2.1)$$

In this case, the weight vector is denoted by w , the input vector by x , and the bias term by b . Points that satisfy this equation define the border, and a point's categorization is indicated by the sign of the equation's result. The kernel approach allows one to project data into a higher-dimensional space in order to identify a linear boundary for data that is not linearly separable. After that, the classification equation is as follows:

$$y = wx + b, y = \{1, -1\} \quad (2.2)$$

SVM algorithms look for the ideal weights (w) and bias (b) in classification jobs to maximize the margin and correctly categorize the training data.

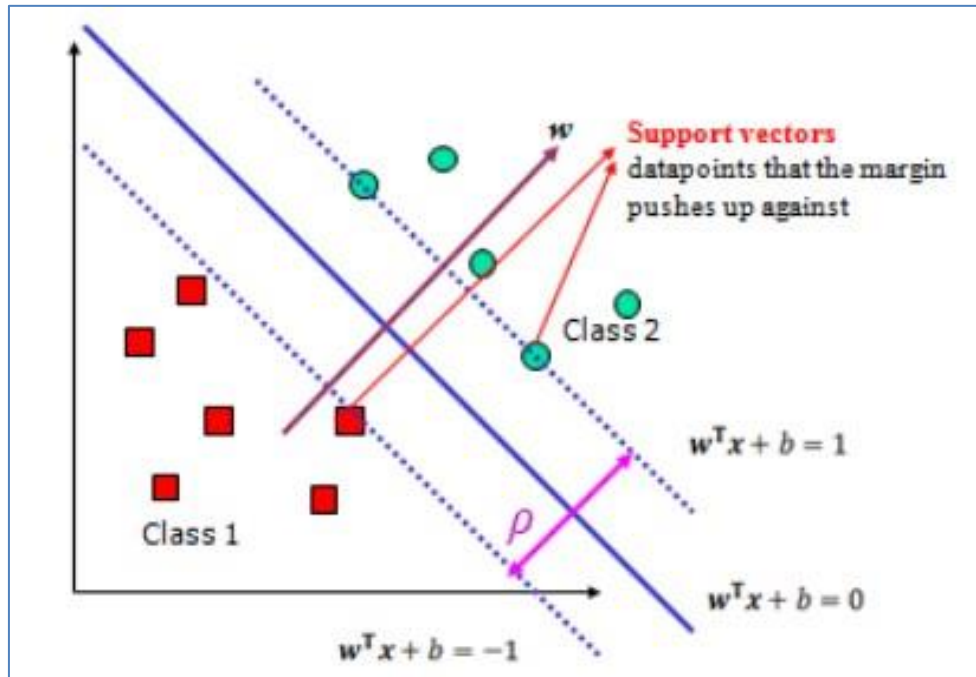


Figure 2.5. Decision boundaries and SVM margin [54].

Gradient Boosting

One popular EL technique for classification and regression applications is gradient boosting (GB) [55]. It works by fusing multiple weak learners together to create a strong learner to make precise predictions. While RF and bootstrapping create weak learners through random feature selection, GB uses gradient descent to reduce the loss function and increase prediction accuracy [56]. Every weak learner in a GB model is specially educated to rectify the mistakes of its predecessor. A composite of all the predictions made by the weak learners makes up the final result. By learning from the mistakes of previous learners, the model can iteratively improve its predicted accuracy through this step-by-step training technique.

The mathematical formula for the Gradient Boosted DT (GBDT) algorithm is represented as:

$$F(x) = F(x) + h(x; \theta) \quad (2.3)$$

Here, $F(x)$ denotes the current model, while $h(x; \theta)$ represents the addition of a new DT to the model, parameterized by θ . The objective is to reduce the loss function $L(y, F(x))$ by fine-tuning the parameters θ of the new DT $h(x; \theta)$. This optimization process is key to enhancing the model's predictive performance.

Artificial Neural Network

Artificial Neural Networks (ANN) draw their conceptual framework from the structure of organic nervous systems [103]. The architecture of an ANN, as illustrated in Fig. 2.5, comprises three fundamental layers:

- **Input Layer:** The size of the input layer corresponds to the dimensions of the training samples in the database. For example, it could equal the number of pixels in an image or words in a sentence [144].
- **Hidden Layer:** The dimensions of hidden layers are not predetermined. Their size needs to be determined experimentally each time the network is tested to ensure it aligns with the accuracy requirements of the network's performance. The hidden layers process the inputs received from the input or preceding hidden layers by multiplying them with their

respective weights. The outcomes of these computations are then forwarded to subsequent hidden layers or the output layer [144].

- **Output Layer:** This layer multiplies inputs by their weights after receiving them from the last hidden layer. The output layer is in charge of generating the network's ultimate choice or output. As such, its structure must correspond with the dimensions of the intended result. For example, the output layer would have six nodes if the job was to categorize phrases into six groups [144].

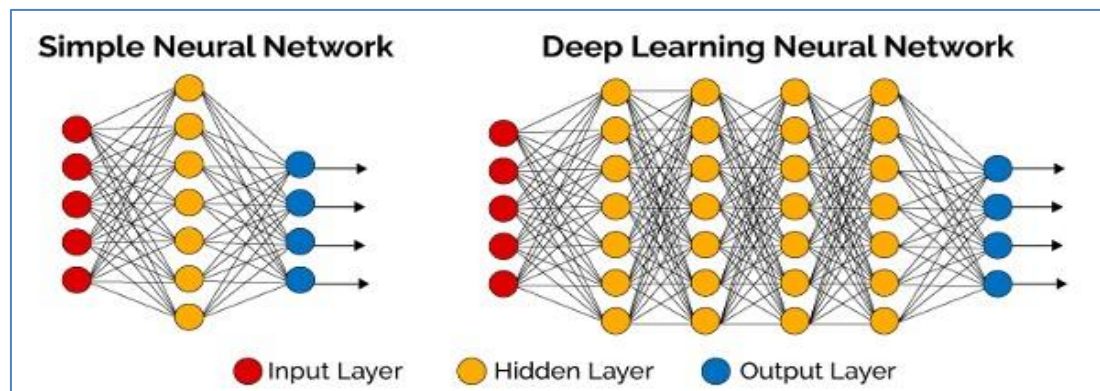


Figure 2.6. Basic layers of ANN.

2.5. Deep Learning

2.5.1. Definition

Deep Learning (DL) employs hierarchical architectures to discern complex patterns in data. This innovative technique has gained widespread application across various fields of AI, such as semantic parsing [58], transfer learning [59], NLP, and computer vision, among others. The current expansion of DL is propelled by three primary factors: a significant increase in chip processing capabilities (GPU units), a substantial reduction in the cost of computing hardware, and notable advancements in ML methodologies.

2.5.2. Convolutional neural networks

CNNs are recognized for their capability to train multiple layers effectively and are widely used in various applications within the field of computer vision [104]. The design workflow of CNNs is depicted in Figure 2.7. Typically, a CNN comprises three main types of neural layers: convolutional layers, pooling layers, and fully connected layers [105]. Figure 2.7

illustrates a standard CNN architecture for image classification, showing the layer-by-layer process. The training of the network involves two main stages: forward and backwards.

During the forward stage, the primary goal is to process the incoming image using each layer's current weights and biases. The loss cost is computed using the ground truth labels and the predicted output. In the subsequent backward stage, gradients for each parameter are calculated based on the loss cost. These gradients are then employed to update all the parameters, preparing them for the following forward computation. The network's learning process can be concluded after a sufficient number of iterations have been completed in both the forward and backward stages. This iterative training process allows for continuously refining the network's performance.

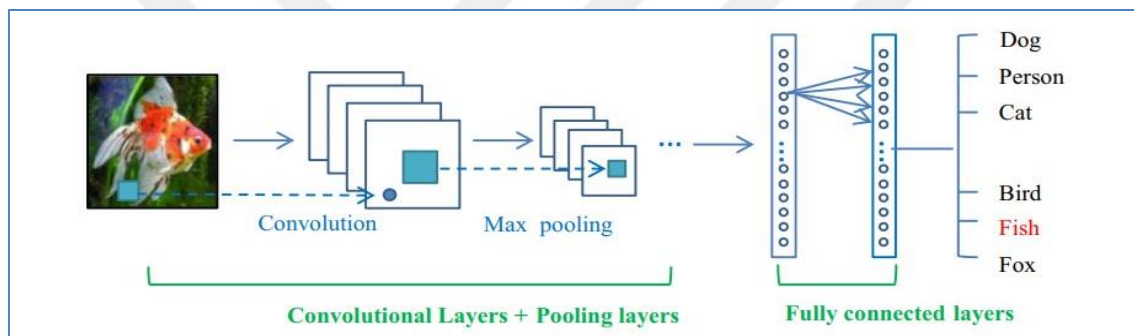


Figure 2.7. Design of CNN

Convolution layer

In a convolution layer, feature extraction from an image is achieved through the dot product of a filter and a kernel defined by an image kernel. Often, a 3x3 kernel matrix is used. Applying a filter to input leads to activation, and although the convolution layer's receptive field is limited, it houses a set of trainable filters maximized for the provided data. Convolution layers are key components of Convolutional Neural Networks (CNNs) [106].

Pooling layer

Following the convolution layer, the pooling layer performs downsampling to reduce the dimensions of feature maps, either by averaging or taking the maximum value in each region. This reduction in dimensions reduces the number of parameters to learn. Two common types of pooling are [107]:

- **Max-Pooling:** Divides the input into smaller sub-regions and extracts the maximum value from each, leading to a condensed but significant input representation.
- **Average-Pooling:** Computes the average value of each sub-region, resulting in a downscaled representation that mitigates the influence of outliers or extreme values.

Activation layer

In Artificial Neural Networks (ANNs), the activation function is a mathematical tool determining a neuron's output [108]. Activation functions can produce a binary output for classification or a continuous value for probability estimation. Common types include the sigmoid, tanh, ReLU functions [109], and softmax functions. The choice of activation function significantly influences the network's performance.

Flatten layer

The flattened layer transforms multi-dimensional input data (like a 2D-pixel grid in an image) into a one-dimensional array [110]. It's often used in CNNs to bridge multi-dimensional convolutional layers and one-dimensional dense layers.

Dense layer

Also known as a fully connected layer, each neuron in a dense layer is connected to every neuron in the preceding layer [110]. Dense layer neurons add non-linearity to the output while learning connection weights by applying a non-linear activation function to the weighted sum of inputs from the preceding layer. Dense layers are typically employed in the final stages of a neural network, post-processing the outputs of convolutional layers. They are crucial for tasks like classification or prediction [111].

Dropout layer

Dropout layers are integral in ANNs to prevent overfitting, a common challenge in ML where models perform well on training data but poorly on unseen data [113]. During training, a dropout layer randomly sets a proportion of outputs from the previous layer to zero, encouraging the network to consider a broader range of input combinations and

reducing reliance on any single neuron. This strategy helps enhance the model's generalization capabilities on new data sets.

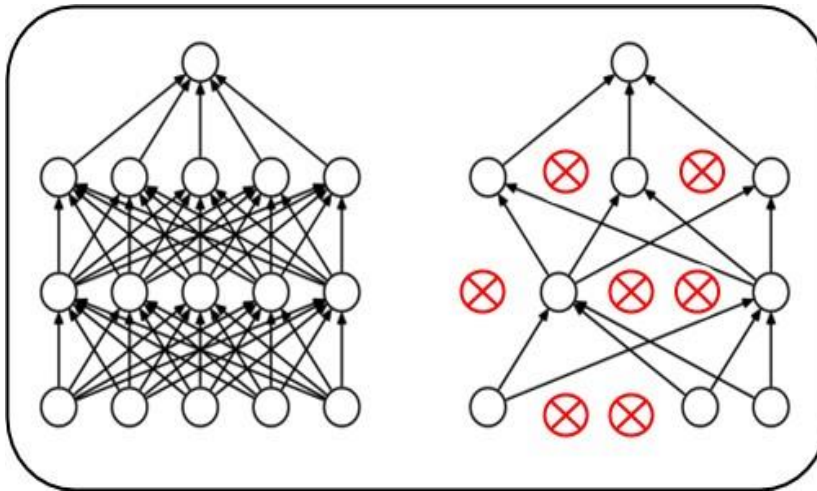


Figure 2.8. Dropout Mechanism [112].

BatchNormalization layer

the Batch Normalization layer is commonly used to expedite training and improve the network's ability to generalize to new data [114]. This layer functions by normalizing the inputs to a network layer during the batch normalization process. It scales the inputs by subtracting the batch mean and dividing by the batch standard deviation, resulting in a mean of zero and a standard deviation of one. This standardization is crucial as it helps maintain the stability of the network by preventing the inputs to a layer from becoming excessively large or small, which could otherwise destabilize the network or halt its learning process.

2.5.3. Recurrent neural networks

ANNs are nonlinear computational models that mimic the architecture and functionality of biological neural networks, such as those found in the brain. Every node or neuron in an ANN uses a particular transfer function, such as sigmoid or Heaviside, to generate an output. These nodes are connected through threshold biases and connection weights. The feedforward network is a primary type of ANN, in which information flows linearly from the input layer to the output layer, bypassing any hidden layers and never looping back. ANNs are particularly good at mapping specific vocabularies onto hidden nodes, which helps them establish semantic linkages between related sentences [115].

In contrast, RNN can use internal states to process sequences of inputs, a feature absent in feedforward networks [116]. A cyclical or feedback link allows RNNs to update their states depending on both current inputs and prior states, setting them apart from feedforward networks. Sequence modelling tasks are one area where RNNs excel because of their recurrent architecture. Retention of knowledge between iterations is one of the main advantages of RNNs, and it is beneficial for NLP tasks involving context generation [117].

RNNs can handle temporally dependent sequences with ease because of their cyclical connections. RNNs can understand the meaning and context of a sequence by employing and storing information from earlier states. Since awareness of the sequential nature of the data is essential for applications like language modelling, machine translation, sentiment analysis, and audio recognition, this feature makes RNNs extremely valuable.

Long short-term memory

LSTM represents an advanced variant of the RNN, known for its effectiveness across a broad spectrum of applications [118]. A distinguishing feature of LSTM is its adeptness at managing long-duration events and delays in time series, making it exceptionally suitable for time series forecasting challenges. Its design allows it to preserve relevant information while discarding the non-essential, addressing the vanishing gradient issue often seen in traditional RNNs. This unique ability of LSTM to recognize and remember long-term dependencies in sequential data enables it to make precise predictions.

Figure 2.9 illustrates the structure of an LSTM model, showcasing its key components and interconnections. The LSTM's defining feature is its memory cell, equipped with a gating mechanism that regulates information flow. This mechanism comprises three principal gates: input, forget, and output gates, each performing a distinct role. These gates allow the LSTM cell to judiciously update and transmit information over time judiciously, thus maintaining relevant data across extended sequences, overcoming traditional RNN limitations, and identifying significant temporal patterns [119].

The Bidirectional LSTM (BiLSTM), an LSTM improvement, considerably improves the efficiency of sequential data processing. BiLSTM generates two unique hidden states by analyzing the forward and backwards input sequences. By merging these states, the network can

incorporate context from before and after the present point in the sequence. This two-way approach improves the model's ability to identify context and dependencies in the data. Many NLP tasks, including text generation, named entity recognition, and sentiment analysis, heavily utilize LSTMs and BiLSTMs. In these applications, their capacity to comprehend data's sequential nature and grasp contextual nuances is essential [120].

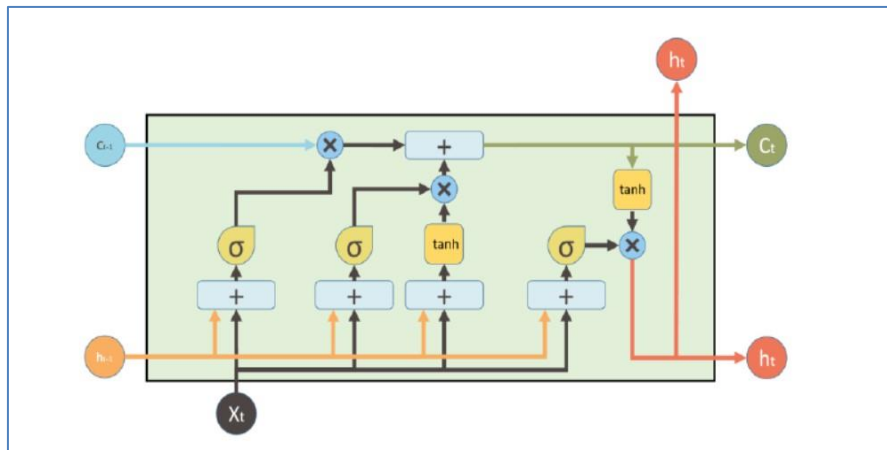


Figure 2.9. Architecture of LSTM [61].

Gated recurrent unit

The GRU is an advanced form of the standard RNN, featuring a gating mechanism akin to the LSTM to overcome limitations associated with short-term memory in RNNs. In GRU, a series of gates is responsible for managing the flow of information and determining what data is essential to keep or discard. This selective information processing enables GRU to effectively learn and make predictions, focusing on pertinent information [121].

Compared to traditional RNNs, GRU's architecture is streamlined yet has enhanced performance and efficiency in various applications. Figure 2.10 displays GRU's fundamental structure, highlighting its crucial elements and interconnections. Through its internal gating system, the GRU cell is adept at learning from and adapting to patterns in sequential data, making it an apt choice for tasks such as NLP, speech recognition, and time series analysis [122].

GRU's gating mechanism aids in retaining significant information over extended sequences, thereby addressing the vanishing gradient issue common in standard RNNs. It expertly navigates the balance between retaining and discarding data, learning to ignore unnecessary details while

holding onto critical information for precise predictions. This trait renders GRU highly effective for ML tasks that demand tracking dependencies over long sequences [122].

GRU is gaining popularity in the scientific community and is praised for its reliable performance and computational efficiency. Because of its less complex design than LSTM, it requires less processing and has fewer parameters, which speeds up training and response times. This makes GRU a desirable choice in scenarios requiring quick, real-time predictions or with limited processing resources.

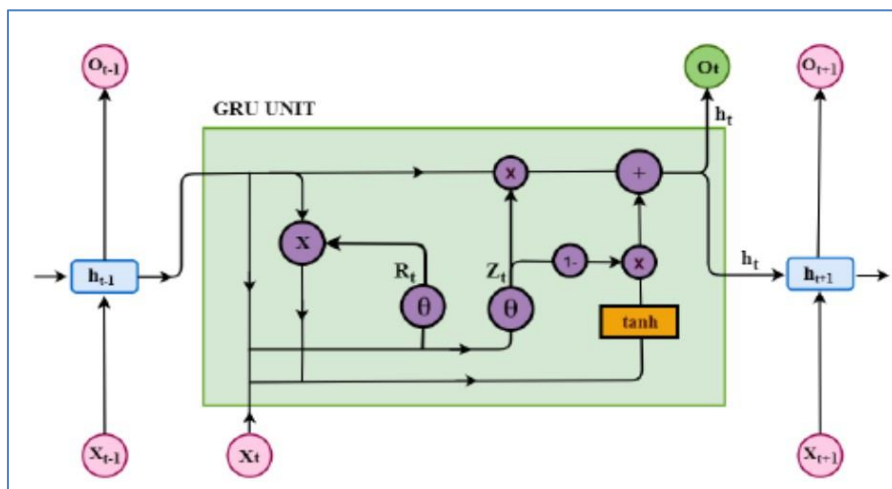


Figure 2.10. Architecture of GAN [62].

2.6. Cross Validation

One statistical technique for evaluating the effectiveness of ML models is cross-validation. Because of its simplicity, usability, and often lesser bias in judging model skills when compared to other approaches, it is widely used in ML [39]. This method splits the dataset into two subsets: the training set and the validation (or test) set. The validation set is used to assess the model's predicted accuracy after it has initially been trained using the training set. This assessment shows how well the model might work with unknown data [63].

A particular kind of cross-validation called "K-Fold cross-validation" divides the dataset into 'k' distinct subsets, or 'folds,' as Figure 4.11 illustrates. The model is tested on the remaining fold after being trained on 'k-1' folds. Every fold serves as the test set once, and this process is repeated 'k' times. Usually, an overall performance statistic is created by averaging or combining the results of each iteration [64].

Compared to techniques like repeated random subsampling, K-fold cross-validation guarantees that each data point is used exactly once for validation and twice for training. This approach is particularly beneficial when dealing with datasets of a reasonable size and when sufficient computational resources are available to train the model 'k' times [65]. This method ensures comprehensive utilization and validation of the data, making it a reliable tool for model assessment.

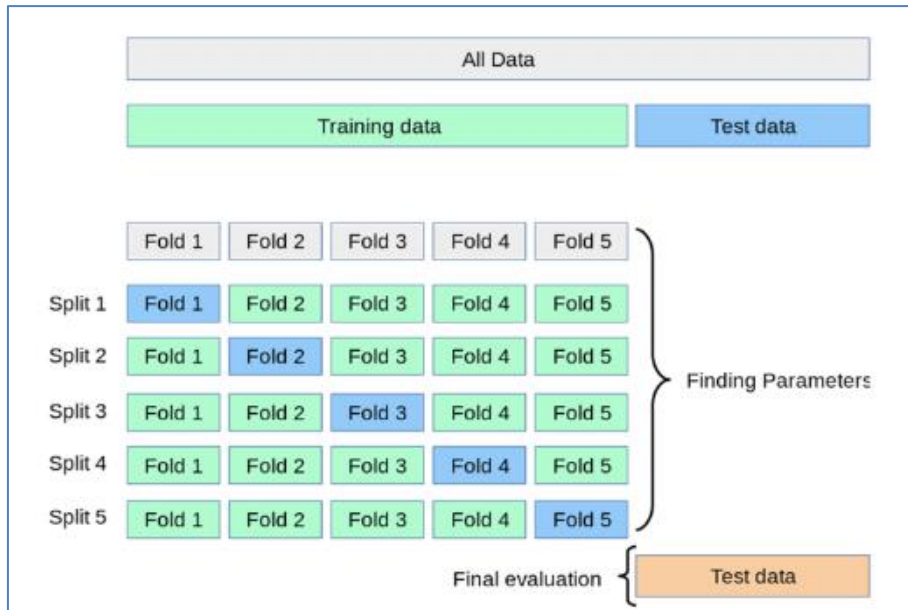


Figure 2.11. K-fold cross validation [66].

2.7. Related Work

The evolving landscape of FND encompasses a variety of methodologies and approaches, preceded by a discussion on the history and significant aspects of fake news utilization. The primary challenge in FND tasks lies in accurately categorizing a text document as either fake news or genuine or assigning it to various ordinal categories. The key hurdle in devising an effective classification system is sourcing an appropriate database.

One study [67] references a set of ensemble ML models to classify news articles. It explores several textual characteristics distinguishable between authentic and fraudulent information. These features are then applied to four real-world datasets to evaluate the effectiveness of the algorithms. A range of ensemble methods are employed to train different ML models using these characteristics, revealing that the proposed ensemble learner outperforms single-

learner methods. Notably, using the ISOT dataset, the RF classifier achieved an impressive ACC of 98%.

A different study [68] examined a number of ML and DL methods, such as character level C-LSTM, convolutional HAN, LSTM, LR, DT, and SVM. Lexical, sentiment, unigram, and bigram techniques with TF-IDF, glove, and character embedding were used to evaluate these models. The findings showed that the LSTM model produced the best results, with the NB approach employing bigram TF-IDF on a mixed corpus dataset obtaining 94% ACC.

Further, a study [69] proposed a FND system using ML techniques, where TF-IDF of a bag of words and n-grams was employed for feature extraction in the pre-processing phase. This approach trained an SVM classifier on a specially curated dataset. In a separate investigation [70], feature extraction was also conducted using the TF-IDF method. This study implemented three ML models: SVM, NB, and Passive Aggressive Classifier, with the SVM classifier attaining the highest ACC of 95.05%.

Two labelled datasets were introduced in the study [71]. The first dataset included real news from reliable sources in six primary categories, compared to fraudulent material from AMTworkers crowdsourcing. The second dataset, which came from internet tabloids, contained real and fake celebrity news. This study examined readability, syntax, psycholinguistic characteristics from the LIWC lexicon, vector space representations of n-grams using the TF-IDF metric, and stylometric aspects such as punctuation from the LIWC lexicon. SVM was the chosen method for detection. The performance varied based on the dataset used; for the first dataset, the best results were achieved using stylometric features, punctuation, and readability, whereas for the second dataset, the most effective classifier combined the complete LIWC lexicon with vector space representation features.

A different study [3] offered an ensemble-based DL model for FND using the LIAR dataset. Two DL models were applied in light of the natural language component of the dataset: a bi-LSTM dense model was utilized for the "statement" features. In contrast, a dense model was applied for all other features. This method demonstrated an ACC of 89% when utilising just the statement property.

Research focusing on detecting fake news in SM [72] employed SVM and RF classifiers, both with and without a 10-fold cross-validation method. The ACC of both classifiers reached 83.5% after applying cross-validation.

A separate study [73] combined DL, NLP, and semantics to create a hybrid method using datasets from LIAR and PolitiFact. This research evaluated the performance of various DL models like Bi-LSTM GRU and CapsNet and traditional ML models, including MNB, SGD, LR, DT, and SVM. CapsNet achieved the highest ACC of 64% on the LIAR dataset. The inclusion of semantic data such as NER and sentiment analysis was found to enhance the performance of the classification model. An ensemble classification model was also developed to accurately identify fake news, using an ensemble of three well-known ML models. This ensemble focused on extracting key features from fake news datasets for classification. The model showed a testing ACC of 44.15% and a training ACC of 99.8% on the LIAR dataset.

In the study "3FND Using DL Techniques" [74], various classifiers were evaluated, including LR, NB, SVM, RF, and DNN. This assessment was conducted on the LIAR dataset, utilizing standard text preprocessing methods from the NLP field, such as stemming and stop word removal.

On the ISOT dataset, 100% training and testing accuracy was reported in a different study [75]. This study used hold-out cross-validation to assess the performance of three DL models and ML on two datasets of genuine and fake news sizes. Methods including word frequency, TF-IDF, and embedding approaches for DL models were used to represent the text data. The authors' unique stacking model produced testing accuracies of 96.05% on the KDnugget dataset and 99.94% on the ISOT dataset.

Another work [76] used ensemble methods to apply vector space representation and stylometric characteristics for text-mining-based FND. Using stylometric data, they created three different feature subsets. The first set included diverse words, complexity measures like the Gunning-Fog index, character counts, and high Flesch-Kincaid reading scores. The second set, derived from a lying detection dataset, featured categories like quantity, vocabulary, grammar, uncertainty, among others. The third subset incorporated the write print feature set for authorship attribution in short texts, covering character, word, syntactic,

structural, and content categories. Various vector space representation techniques were employed, including BoW, TF-IDF, continuous CBoW, and skip-gram (SG) using Word2Vec and FastText tools. Feature selection for stylometric and word vector space features involved recursive reduction and Chi-square testing for lemmatization and stemming. The GB with CBoW Word2Vec embeddings outperformed all other ML techniques, with CBoW representation notably enhancing the performance of non-ensemble algorithms.

2.8. Conclusion

This chapter provides an overview of the essential terminology relevant to our research in FND. Furthermore, it highlights the latest related works documented in the literature. The subsequent chapter will detail our proposed methodology, designed to fulfil the objectives of this research.

3. METHODOLOGY

3.1. Introduction

The escalating issue of widespread misinformation, mainly through fake news, underscores the necessity for efficient and reliable detection methods to identify deceptive content autonomously. In this context, the methodology chapter is critical to our research. It elaborately describes the strategies and approaches to address and analyze this intricate challenge. This chapter combines academic rigour and real-world relevance by taking the reader through our research methodology and explaining the thinking and techniques that went into each choice.

Our methodology is organized into several key phases, each contributing to a holistic understanding of FND. The process begins with selecting a suitable dataset, underscoring its pivotal role as the groundwork for the following research stages. We then examine the significance of data preprocessing, explaining how the raw dataset undergoes cleansing, refinement, and transformation, preparing it for practical model training.

Our work is primarily concerned with the application of DL models. We discuss the architecture and selection criteria for the CNN, CNN-LSTM, and CNN-GRU architectural combinations. The selection of layers, filter size, and activation functions are given particular consideration to guarantee a thorough and trustworthy analysis. The testing and training procedures are then covered, utilizing industry best practices to guarantee our models' efficiency and broad applicability. The final section of the chapter is devoted to describing the performance indicators we have selected to use in our model assessment.

3.2. Proposed Approach

Figure 3.1 demonstrates the structured process involved in FND, comprising several sequential phases. The initial step involves gathering data, specifically using the ISOT dataset, recognized for its relevance in this field. After data acquisition, an extensive preprocessing phase is executed. This phase focuses on cleaning and refining the data to ensure its readiness for later stages.

After preprocessing, the focus shifts to deploying our proposed classification models. These models, intricately designed, are pivotal in identifying fake news. Following their setup, these classifiers enter a training phase where they assimilate knowledge from the processed dataset. This training is essential for fine-tuning the models' parameters and boosting their ability to predict accurately.

Once training is complete, the models undergo a testing phase, utilizing distinct subsets of the same refined dataset for validation. This step is critical for evaluating their generalization skills and overall performance. The concluding phase involves an in-depth analysis to assess the efficacy of each model. This evaluation includes a range of performance metrics, as previously outlined, offering a thorough insight into the strengths and limitations of each model. Additionally, comparative analyses are conducted to discern the performance variations among different models.

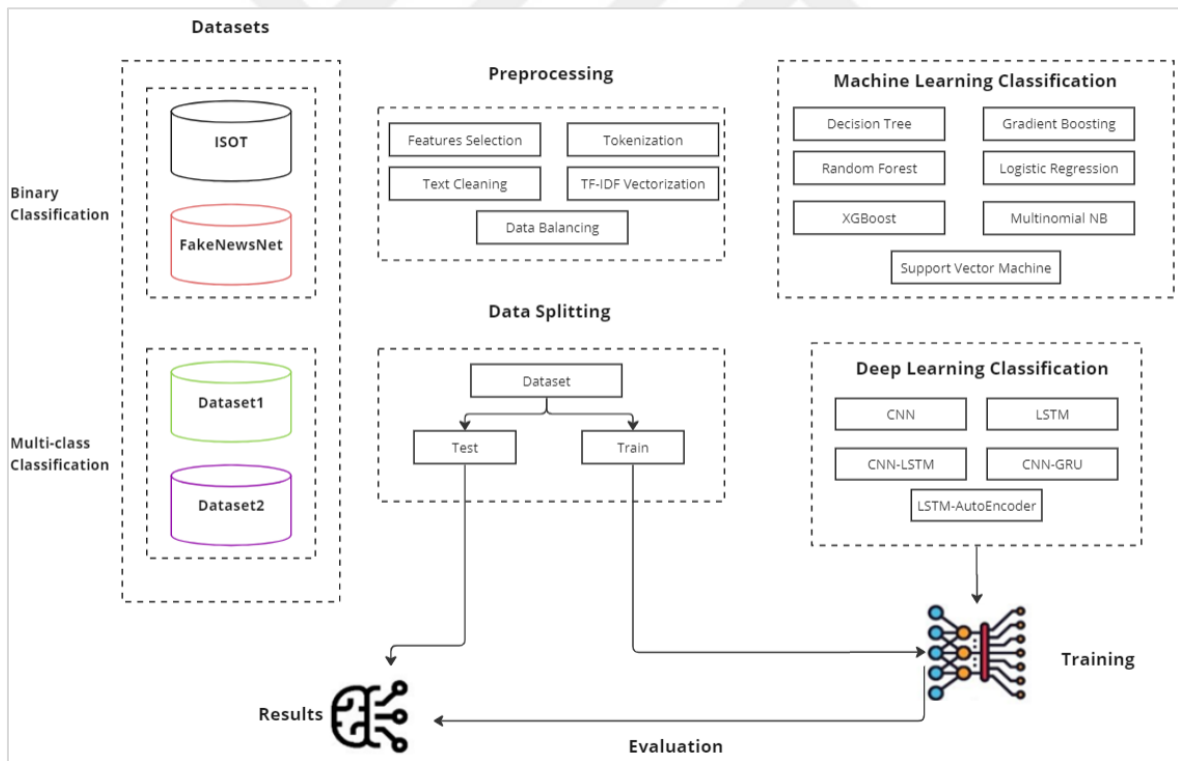


Figure 3.1. FND proposed framework

3.2.1. Datasets description

To effectively identify and categorize fake news, it's imperative to have a comprehensive dataset that encapsulates the diverse aspects of truthfulness in news reporting. Consequently,

our methodology amalgamates four distinct datasets, facilitating a more nuanced and thorough examination.

The ISOT Corpus:

In our analysis, we utilize the ISOT corpus, a carefully assembled dataset tailored for binary classification of news articles into "fake" or "real" categories derived from authentic sources [77]. This corpus includes 44,898 articles, with 21,417 categorized as real and 23,481 as fake, divided into two separate files: "True.csv" and "Fake.csv", each containing over 12,600 articles.

- **Real News Articles:** The "True.csv" file comprises articles exclusively from Reuters.com, renowned for their reliability, predominantly focusing on politics and global events. These articles, mostly gathered between 2016 and 2017, contain metadata elements, including title, text, news type, and publication date.
- **Fake News Articles:** Conversely, the "Fake.csv" file aggregates articles from various untrustworthy websites, some of which have been flagged by fact-checking organizations like Politifact and Wikipedia. These articles, mirroring the real news, span various subjects, especially politics and international affairs. Notably, the dataset retains any grammatical or punctuation peculiarities in these articles to preserve the originality and integrity of the data.

FakeNewsNet Dataset:

A specialized resource created for scholars exploring the intricacies of fake news is the FakeNewsNet dataset. It is founded on verified truths from two distinct and credible sources, Politifact and Gossipcop, offering a comprehensive perspective for studying the nuances of fake news [78]-[80].

The FakeNewsNet dataset, consisting of 23,196 rows and five essential columns, is organized to offer comprehensive understanding of the dynamics involved in disseminating fake news. Its design is tailored for researcher accessibility and efficiency, incorporating five essential attributes:

- **ID:** Each record is assigned a unique identifier, simplifying referencing and handling individual data points.

- **News_URL:** The dataset includes direct links to the original news articles, facilitating validation and in-depth source content analysis.
- **Title:** The title or headline of the news piece is listed, providing a snapshot of the article's content and acting as the initial point of engagement for readers.
- **Tweet_IDs:** This column aggregates the Twitter IDs linked to tweets that have shared the news article, offering insights into the article's distribution on SM and its broader impact.
- **Label:** Employing a binary classification, the dataset labels each article as either 'Fake' or 'Real'. This classification is crucial for applying supervised learning techniques or conducting comparative studies in fake news detection.

The FakeNewsNet dataset is a well-organized and rich resource, apt for intricate analyses in the evolving domain of FND and understanding.

The LIAR-PLUS Dataset:

Incorporating elements from the LIAR-PLUS dataset [81] significantly enhances our methodology for FND and classification. Including 12,836 short statements that have been manually annotated according to standards such as truthfulness, subject matter, and context, this dataset is an extension of the original LIAR dataset, which was first presented by Wang in 2017. In the dataset, truthfulness is categorized along a spectrum from "pants-fire" to "true."

The LIAR-PLUS dataset has been improved significantly by adding automatically extracted reasons or evidence sentences from full-text verdict reports written by Politifact journalists. Usually found in summary sections with titles like "our ruling" or "summing up," these explanations provide context for the honesty grade that has been assigned. If there isn't a summary section, the final five sentences of the fact-checking article are utilized instead of any that state the outcome explicitly.

The LIAR-PLUS dataset is comprised of 15 columns, presenting a comprehensive multi-dimensional perspective:

- **ID:** A special number is assigned to each assessed statement.
- **Label:** A sincerity indicator ranging from "pants-fire" to "true."

- Statement: The assertion being investigated.
- Subjects: The topic the statement addresses includes the health care system and the economy.
- Speaker Details: Specifics about the individual delivering the speech.
- State Information: Any pertinent geographic context.
- Party Affiliation: The speaker's political affiliation.
- The credit history of the speaker is evaluated based on their previous ratings of truthfulness, which are categorized as "barely true," "false," "half true," "mostly true," and "pants on fire."
- Context: The platform or environment in which the statement was delivered, such as a tweet or a TV interview.
- Excerpted Justification: Verification sentences from articles that verify or refute the claim.

Our goal in integrating the LIAR-PLUS dataset is to improve the quality of evidence retrieval, which will help our FND algorithms perform much better. We hope to strengthen the categorization process by employing these justification sentences to provide the contextual depth necessary for determining the integrity of news pieces.

Combining the datasets:

Through careful integration of the ISOT corpus with the LIAR-PLUS dataset, our research on FND represents a substantial development. Beyond simple binary categorization, this fusion allows for a more nuanced approach to FND, incorporating categories like "Half-True" and "Barely True." We then combine the ISOT corpus with the "Half-True" category from LIAR-PLUS to create our first dataset, which we label as Dataset 1. To build on this, we have another dataset called Dataset 2, which includes the "Barely True" category and makes Dataset 1 larger. A composite dataset is created by combining the ISOT corpus with "Half-True" and "Barely True" components from LIAR-PLUS.

This layered and hierarchical assembly of data is designed to enhance the complexity and depth of our classification method progressively. By meticulously blending components from both datasets, our model is adept at addressing the multifaceted challenges of fake news. It transcends the identification of outright fabrications, enabling the detection of more

nuanced forms of misinformation. Our approach, rooted in recognizing varying degrees of truthfulness and falsity, underscores the need for sophisticated analysis in the contemporary digital milieu. Consequently, we present a refined and potent analytical framework that strengthens our collective ability to combat the spread of fake news.

3.2.2. Preprocessing

The efficacy of ML models is heavily influenced by the quality of the input data, necessitating comprehensive pre-processing steps to optimize the data for practical analysis. Our pre-processing pipeline involves a series of techniques, each aimed at enhancing the model's capacity to identify key terms and thereby boost its performance. The steps in our pre-processing regimen include:

Feature Selection:

We start by concentrating on the "title" and "text" features from the ISOT corpus and LIAR-PLUS, our main data sources. We achieve distribution balance for the FakeNewsNet dataset by downsampling the "real" news articles to match the "fake" news volume. Our primary feature set for every dataset is created by combining these two carefully chosen features, title and text, into a single column.

Text Cleaning:

The next step involves changing all text to lowercase and using regular expressions to eliminate HTML tags, punctuation, and unnecessary spaces. This standardization guarantees consistency throughout the dataset and aids in simplifying it.

Tokenization:

Tokenization, a key technique in Natural Language Processing (NLP), is the third step. It breaks down the cleaned text into smaller units or tokens, such as words or phrases. This process simplifies the text into a list of fundamental elements, setting the stage for feature extraction.

Term Frequency-Inverse Document Frequency (TF-IDF):

We use TF-IDF, a statistical technique to assess a word's importance in a document about a corpus [82]. Words important in a particular document but uncommon throughout the corpus are highlighted by the TF-IDF score, which is determined by multiplying term frequency (TF) by inverse document frequency (IDF). This makes them valuable differentiators.

Data Splitting:

Following extensive pre-processing, the dataset was wisely divided into various subsets for training and testing our prediction model in our academic research. We specifically selected 80% of the dataset for training, with the remaining 20% as the test set. This division was rigorously designed to provide a strong sample size for the training phase while enabling an unbiased assessment of the model's performance during the testing phase. In addition, we used a stratified K-Fold cross-validation approach using five folds to improve the reliability of our model evaluation. This method increases the value of the training corpus by allowing each datum to evaluate the model and maintains the proportionality of the classes across each fold. Combining these strategies enables our model to recognize and assess the various and complex expressions of misinformation that affect the digital news realm with greater accuracy and precision.

3.2.3. Machine learning based classifiers

We have applied eight different ML algorithms in our study to identify fake news in the ISOT dataset efficiently. Here, we examine each algorithm, emphasizing its core ideas and capabilities.

Decision Tree (DT)

The DT algorithm is a cornerstone in ML, renowned for its straightforward approach in classification and regression tasks [83]. At its core, a DT iteratively divides a dataset into increasingly finer subsets while concurrently constructing an associated DT. The final model is a tree comprising decision and leaf nodes. Decision nodes represent the different possible values for an input feature, while leaf nodes signify the final decision or output after considering all inputs.

In terms of architecture, DT uses binary recursive partitioning to produce hierarchical models. They are created top-down, breaking the data into segments using the best feature at each stage. Measures of impurity, such as the mean squared error for regression trees and the Gini index for classification trees, form the basis of this division. These measurements are based on the following mathematical formulas:

$$H(S) = - \sum p(x) \log_2 p(x)$$

for entropy, and

$$G(S) = 1 - \sum [p(x)]^2$$

For Gini impurity, where 'S' is a dataset and 'p(x)' is the probability of class 'x'.

The DT grows recursively, forming a deep structure until it reaches a pre-set depth or a minimum number of samples, avoiding overfitting. Information gain, a key concept in DTs, is calculated by:

$$IG(S, A) = H(S) - \sum \left(\frac{|S_v|}{|S|} \cdot H(S_v) \right)$$

Where 'IG(S, A)' represents the information gained from splitting dataset 'S' based on attribute 'A'.

Pruning techniques like cost-complexity pruning are used to prevent overfitting by eliminating less influential branches. This integrates a parameter 'alpha' to balance tree size and fit. Additionally, hyperparameters such as tree depth, leaf size, and the number of features per split are fine-tuned, balancing underfitting and overfitting.

DTs are known for their interpretability and simplicity, handling numerical and categorical data. They apply in various fields, from NLP to financial modelling. However, they can be sensitive to data noise and struggle with smooth decision boundaries, issues that ensemble methods like RFs and GB can address. These methods enhance robustness by combining the

strengths of multiple DTs, underscoring the versatility and foundational role of DTs in the evolving landscape of ML.

Random Forest (RF)

The RF technique is a fundamental component of EL, utilizing numerous DTs' combined strength and robustness to build a well-known model for these qualities [84]. The approach is based on the EL theory, which suggests that multiple weak learners can be combined to create a single strong learner, improving the predictive power and stability of the model. The central limit theorem and the law of large numbers are two key mathematical and probabilistic ideas that underpin this idea. According to these ideas, merging a sizable number of relatively independent trees converges towards a normal distribution, increasing prediction accuracy and stabilizing results.

During the training phase of the RF algorithm, numerous DTs are developed through a stochastic process that selects both features and data points, an approach based on bagging. Each tree in the ensemble independently predicts new data, with the diversity among trees ensured by selecting random subsets of features at each node, a method reliant on combinatorial and optimization techniques. The bagging process, which generates different bootstrap samples for training individual trees, aims to reduce prediction variance while slightly increasing bias. The final prediction in RF is either the average of all tree predictions for regression or the majority vote for classification, typically resulting in improved generalization error. This error is examined using bias-variance decomposition, which is crucial for understanding the balance between model complexity and predictive accuracy.

In RF, decision theory guides the output class determination in classification tasks through a majority vote, mathematically represented as:

$$C_f(X) = \text{mode}\{C_1(X), C_2(X), \dots, C_T(X)\}$$

Where T is the number of trees $C_f(X)$ is the output class of tree t , and $Y_f(X)$ is the output of tree t . The mean forecast made by every tree is the result of regression:

$$Y_f(X) = \frac{1}{T} \sum_{t=1}^T Y_t(X)$$

Furthermore, RF offers analytical insights through feature importance analysis, which reveals each feature's contribution to the predictive process, typically based on the reduction in impurity caused by feature-associated splits across all trees.

By blending mathematical principles with practical application, RF balances bias and variance, integrating numerous simple models into a more sophisticated predictive tool. Its capacity for parallelized tree construction enhances scalability, which is particularly beneficial in significant data contexts. Thus, the RF algorithm remains a vital tool in modern ML, exemplifying the effectiveness of combining diverse predictive models for optimal results.

Gradient Boosting

By methodically combining weaker learning models, especially DT, the GB algorithm is an advanced EL technique that is based on mathematical and computational concepts and allows the development of powerful predictive models [85].

The foundation of GB is the boosting approach, an ensemble technique that builds a robust classifier by combining several weak classifiers (models that perform just marginally better than random guessing). This approach uses a step-by-step additive strategy that iteratively improves predictions as part of a sequential learning process. During each iteration, the errors remaining from previous models guide the development of new models aimed at rectifying these errors [85]. Gradient descent optimization, a basic method for locating the function that minimizes a differentiable loss function inside the function space, is used to coordinate this error correction. Mathematically, it is expressed as:

$$F(x) = \sum_{i=1}^M \alpha_i h_i(x)$$

Where, $F(x)$ is the final predictive model, M is the number of trees, α_i is the weight given to the i -th tree, and h_i is the prediction of the i -th tree.

The method of gradient descent is used to guide the system repeatedly towards minimizing the loss function, with each tree in the series seeking to improve upon the errors of its predecessors in this organized way. A number of factors, such as the learning rate, which affects the contribution of each tree, serve as control mechanisms to ward off overfitting and encourage the creation of more comprehensive models [85].

The method of gradient descent is used to guide the system repeatedly towards minimizing the loss function, with each tree in the series seeking to improve upon the errors of its predecessors in this organized way. A number of factors, such as the learning rate, which affects the contribution of each tree, serve as control mechanisms to ward off overfitting and encourage the creation of more comprehensive models [85]. With careful construction, this method yields a layered, networked predictive system that improves accuracy over time by learning from past mistakes. The mathematical framework produces a flexible learning environment capable of optimizing intricate loss functions and managing a variety of data kinds [85].

The GB technique is an intricate instrument based on ideas related to statistics, optimisation, and computing that collaborate to create exceptionally exact and accurate predictive models. It is an important component of contemporary ML, demonstrating exceptional function space optimization and complexity adaptation across various prediction scenarios. This makes it an essential tool in today's ML toolbox, able to handle complex predictive modeling difficulties accurately and precisely [85].

Logistic Regression (LR)

A key component of both ML and statistics, LR is a flexible and fundamental supervised learning technique [86]. When there are two potential categories for the outcome variable, it is frequently used in binary classification issues. The sigmoid function, or logistic function, is the central component of this algorithm and is defined as $f(x) = \frac{1}{1+e^{-x}}$.

This function accepts a real number as input and returns a value between 0 and 1, which indicates the likelihood that a given input falls into a given category.

Mathematically speaking, the LR model postulates a linear relationship between the independent variables and the log-odds of an occurrence. The logit function is explicitly used to express this relationship as $\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$, where p is the positive class probability. A linear relationship between the independent variables and the log-odds is established by leveraging the logit function, as shown below:

$$\text{logit}(p(Y = 1|X)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$$

The dependent variable, Y , the feature vectors, and the estimated parameters, β , are all represented in this equation. Maximizing the likelihood function, which expresses the probability of seeing the provided dataset, the maximum likelihood estimation approach is utilized to ascertain the ideal parameter values [86].

Regularization strategies such as ridge and lasso regularization are frequently used to avoid overfitting in real-world LR applications. These techniques promote simpler models that perform well in generalizing to new, unseen data by limiting the magnitudes of the weights assigned to each feature. Furthermore, meticulous feature engineering and selection are essential to the effectiveness of LR models. Additionally, LR has an easily understood architecture from a computational perspective, which makes it a good option when the interpretability of the model takes precedence over prediction accuracy. Even though it is straightforward, LR is nevertheless an effective tool, particularly when paired with other methods like polynomial regression to add non-linearity to the model and improve its predictive power [86].

XGBoost

XGBoost is a sophisticated GB DT implementation designed for remarkable speed and efficiency [87]. In order to produce a strong learner who can make accurate predictions, this method puts together a group of weak learners, usually in the form of DTs. It uses the GB technique, which is based on optimizing a differentiable loss function by adding weak learners that correct the mistakes made by the ensemble's previous members.

The objective function of XGBoost, which consists of a regularization term and a particular loss function, can be used to explain the mathematical basis of the algorithm. The algorithm

is directed toward producing the best accurate predictions via the loss function, which measures the difference between expected and actual values [87]. On the other hand, the regularization term helps to prevent overfitting by regulating the model's complexity. The objective function has the following formal definition:

$$\text{Obj}(\Theta) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{i=1}^t \Omega(f_i)$$

Where, $\text{Obj}(\Theta)$ is the objective function, $l(y_i, \hat{y}_i)$ is the loss function, $\Omega(f_i)$ is the regularization term, y_i is the true label, for instance, i , $\hat{y}_i$ is the predicted label, for instance, i , f_i is the i -th tree, and t is the number of trees

XGBoost uses a greedy technique to find the best split points for the DTs, approximating the loss function with a second-order Taylor expansion. This helps determine which structure is best for each unique tree. This iterative procedure aims to create a new tree that reduces the total loss function at each iteration [87].

The two main components of the XGBoost algorithm that increase its speed and effectiveness are parallelization and tree pruning. Multiple computations can be carried out concurrently thanks to parallelization, which greatly shortens training times. On the other hand, XGBoost uses the depth-first method for tree pruning, which regulates the size of DTs [87].

A variety of hyperparameters provided by XGBoost enable users to control the algorithm's behaviour. Important hyperparameters include `max_depth`, which sets the maximum DT depth, and `learning rate`, which controls the step size in gradient descent optimization. To get the best-predicted performance out of XGBoost, fine-tuning these hyperparameters is essential [87].

The XGBoost algorithm is a powerful tool for ML professionals, offering remarkable prediction performance and accuracy by combining sophisticated optimization methods with effective regularization and computational tactics. It is solidly based on a mathematical foundation [87].

Multinomial Naïve Bayes

Multinomial Naïve Bayes (MNB) is a probabilistic learning method commonly used in NLP [145]. As a variant of the NB classifier, MNB is based on Bayes' theorem with the “naïve” assumption of conditional independence between every pair of features given the value of the class variable. This model is particularly suited for classification tasks where features are represented as counts or frequency counts, thus lending itself well to text classification problems where the features are typically the word counts or normalized word frequencies within the documents [146].

Bayes' theorem, in the context of classification, can be expressed as:

$$P(C_k | x_1, \dots, x_n) = \frac{P(C_k) \cdot P(x_1, \dots, x_n | C_k)}{P(x_1, \dots, x_n)}$$

Here, C_k represents a class variable, and $x_1, \dots, x_n$ represent feature vectors. In MNB, the term $P(x_1, \dots, x_n | C_k)$ is computed as:

$$P(x_1, \dots, x_n | C_k) = \prod_{i=1}^n P(x_i | C_k)$$

The model parameters are estimated by Maximum A Posteriori (MAP) estimation, which computes the probability of a word w given a class, C_k . As:

$$P(w | C_k) = \frac{N_{w, C_k} + \alpha}{N_{C_k} + \alpha n}$$

Where:

- N_{w, C_k} Is the number of times word w appears in documents of class C_k ,
- N_{C_k} is the total count of all words in the documents of the class C_k
- n is the number of features (vocabulary size),
- and α is the smoothing parameter (often set to 1, corresponding to Laplace smoothing).

The denominator in the MNB equation ensures that the probabilities are normalized, summing up to 1. When a new document needs to be classified, the MNB algorithm

calculates the posterior probability for each class and assigns the document to the class with the highest posterior probability.

The independence assumption simplifies computation and allows each feature to contribute independently to the probability, making it a computationally efficient algorithm, especially for high-dimensional datasets. However, this assumption is also a limitation because it often does not hold in real-world scenarios where features can be interdependent.

Despite its simplicity and the sometimes overly strong independence assumption, MNB often performs surprisingly well on complex tasks, especially in domains where the features are multinomially distributed, making it a staple algorithm in the text classification toolkit.

Support Vector Machine (SVM)

The SVM is a popular supervised ML technique for problems and outlier detection [88]. The underlying idea behind SVM is finding the ideal hyperplane to efficiently divide data points into different classes within a given feature space. This hyperplane is positioned to maximize the margin between the nearest data points, known as support vectors belonging to different classes in binary classification scenarios. From a mathematical perspective, SVM aims to solve an optimization problem defined by the following primal formulation:

$$\begin{aligned} & \text{minimize} && \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ & \text{subject to} && y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, \dots, n \\ & && \xi_i \geq 0, i = 1, \dots, n \end{aligned}$$

Where $\mathbf{x}_i$ stands for individual data points, y_i for the label associated with $\mathbf{x}_i$, b for the bias term, C for the regularization parameter controlling the trade-off between maximizing the margin and minimizing the classification error, and ξ_i for the slack variables that enable soft-margin classification, making it easier to handle non-linearly separable data and preventing overfitting.

SVM uses a kernel technique, which maps input features into a higher-dimensional space where a hyperplane may successfully separate the classes to increase its application beyond linearly separable issues. RBF, polynomial, and linear kernels are often utilized kernels. The

choice of kernel and parameter tuning are crucial phases in deploying SVM, as they greatly impact the machine's performance [88].

SVM performs exceptionally well in high-dimensional spaces in real life, offering a reliable solution to classification issues involving many features. Utilizing the structural risk minimization concept, which attempts to reduce the upper bound of generalization error rather than empirical error, SVM is also well-known for producing models with strong generalization performance [88].

Despite their resilience and rigour in mathematics, SVMs have limitations, especially regarding scalability for large datasets because of the computing cost of the quadratic programming issue during training. However, SVM continues to be a powerful tool in the ML toolbox, able to reveal significant patterns and draw complex boundaries in a variety of data environments, given the right optimizations and kernel choices. This emphasizes its importance for industrial and scholarly uses [88].

3.2.4. Deep learning based classifiers

Since the challenges under examination are complex, we carefully build various model configurations. In this work, we thoroughly investigate the advanced capabilities of DL, concentrating on two basic architectural paradigms: CNNs and RNNs. These include CNN-GRU and CNN-LSTM hybrid architectures and standalone CNN and RNN models. Every model is made to address specific scientific questions and computational challenges specifically.

Convolutional Neural Network (CNN)

One popular type of DNN used for visual imaging processing is the CNN. Automatic and adaptive learning of hierarchical spatial features from input data is the foundation of its architecture. With ever more complicated layers of neurons identifying patterns in the input data, CNNs are designed to mimic the information processing of a visual cortex [123]. Higher-level characteristics are gradually extracted from the input data by a stack of discrete layers in a standard CNN architecture. These strata consist of :

- Input layer: the first layer, the image is input into the network.

- Convolutional layers: These layers employ filters (kernels) to analyze the input data. Different features, such as edges and corners, are isolated through convolution. Each filter specializes in detecting specific features.
- Activation layers: Non-linear activation functions, such as Rectified Linear Units (ReLU), induce non-linearity after convolution so that the network can learn from mistakes.
- Pooling layers minimize the input's spatial dimensions (width and height), lowering memory consumption and processing stress. It is usual practice to use max-pooling, which takes the most significant value across a range of values in the feature map.
- Fully connected layers: These layers classify the extracted features for pattern recognition, predictions, or classifications by connecting the neurons to the flattened output of the previous layers.
- Output layer: The last layer uses learned high-level features to generate the network's output.
- Normalization and dropout layers: These can be added to enhance performance, avoid overfitting, and facilitate quicker convergence.

Backpropagation algorithms and optimization strategies like gradient descent are used in CNN training. These allow the network to discover the ideal weights, or parameters, that reduce the training data's error between predicted and actual labels [123].

The convolutional layer uses the convolution operation, multiplying a filter element by an element and a localized portion of the input before adding the results into a single output pixel. This process creates an output map that highlights features that are found by scanning the total input [123].

We sequentially operate with this model, beginning with an embedding layer that converts input sequence words into high-dimensional vectors of 128 units. For higher-level processing, this offers a dense representation of the input. Dropout layers are used to stop overfitting, which happens to many models trained on small amounts of data [123].

Next, localized patterns in the data are captured by three different convolutional layers, each with 128 filters. A large range of spatial feature extraction is ensured by varying the kernel sizes (3, 4, and 5). By reducing the dimensionality of the internal representation of the model,

a global max-pooling layer preserves essential features for classification. Ultimately, these traits are mapped by one or more dense layers to a space where intricate correlations can be discovered. A dense layer that is activated by a sigmoid function at the end of the model generates classification probabilities for every class label [123].

LSTM

LSTM networks are a specialized Recurrent Neural Network (RNN) architecture meticulously crafted to address the challenges encountered in training traditional RNNs. These challenges primarily stem from the problems of vanishing and exploding gradients, which hinder the ability to capture and learn long-range dependencies within sequences. LSTMs are purposefully designed to overcome these obstacles by incorporating a set of gating mechanisms that control the flow of information [147].

The architectural complexity of an LSTM unit is characterized by three distinct gates: the input gate, the forget gate, and the output gate. Each gate within an LSTM cell operates like a sentinel, determining the degree of influence a new input should have, the extent to which existing information should be retained, and the proportion of internal states to be passed to the next time step, respectively [119].

The input gate manages the influx of new information into the cell state by applying a sigmoid function to weigh the input and scaling it appropriately through point-wise multiplication with the tanh of the input. Simultaneously, the forget gate employs its own sigmoid function to decide which information should be discarded from the cell state, considering both the previous and current input. This process results in an update to the cell state, a refined combination of the old state, modified by the forget gate, and the new candidate values filtered through the input gate [148].

Following the cell state update, the output gate plays a crucial role. It determines the next hidden state, which serves as the LSTM's output for the current time step and contributes to the final network output. This determination involves applying a sigmoid function to the current input and the previous hidden state, thereby deciding which parts of the cell state should be output. Subsequently, the cell state undergoes a tanh function (introducing non-linearity) and is point-wise multiplied by the output of the sigmoid gate, resulting in the final

output. Some variations of LSTM further enhance the architecture with peephole connections, providing the gates with a direct view of the cell state and improving the model's ability to learn the precise timing of outputs [149].

The practical implications of LSTMs are profound, as they excel in a wide range of tasks, including language modelling, translation, sequence prediction, and more. The intricate interplay of gates and states within the LSTM allows it to remember information over extended periods and selectively forget irrelevant data. Consequently, LSTMs represent a significant advancement in the field of neural networks, enabling the discovery and learning of patterns in sequential data that were previously considered beyond the capabilities of machine understanding [150].

AutoEncoder

Autoencoders represent a neural network paradigm that embodies the pursuit of dimensionality reduction and feature learning. The primary objective of an autoencoder is to learn a representation (encoding) for a given dataset, typically for dimensionality reduction purposes, by incorporating a bottleneck within the network that enforces a compressed knowledge representation of the original input [151].

The architecture of an autoencoder exhibits elegant symmetry, comprising two main components: the encoder and the decoder. The encoder's role is to compress the input data into a latent-space representation. It achieves this by progressively reducing the dimensionality of the input data through various layers, ultimately reaching the bottleneck, which contains the encoded (compressed) information. In contrast, the decoder, mirroring the encoder, takes this encoded input and reconstructs the data to its original dimensionality [152].

The effectiveness of an autoencoder is assessed by its ability to reconstruct the original input from the compressed code accurately. During training, the network adjusts its weights to minimize the reconstruction loss, often quantified using a function like mean squared error. Through this training process, the network learns to emphasize the most relevant attributes of the input data. Interestingly, various variations of autoencoders, including sparse, denoising, and variational autoencoders, introduce additional constraints or objectives to the

fundamental architecture, enhancing robustness and feature learning. For instance, sparse autoencoders employ regularization techniques to enforce sparsity in the encoded representations, encouraging the network to capture more distinctive features.

On the other hand, denoising autoencoders are trained to eliminate noise from the input, resulting in a more resilient data representation. Variational autoencoders have gained significant attention and introduced a probabilistic element into the encoding process by modelling the latent space with a probability distribution. This enables the generation of new instances similar to the input data [153].

Autoencoders' architectural structure and training approach make them highly versatile, suitable for data denoising, generative modelling, and even as a pre-training step for other neural networks. As we navigate the data-driven era, the significance of autoencoders continues to grow, providing valuable insights into the inherent structures and patterns concealed within complex datasets.

LSTM-AutoEncoder

In advanced neural network research, the hybrid LSTM-Autoencoder architecture has emerged as a sophisticated model that harnesses the strengths of both LSTM and Autoencoders. This composite structure is expertly designed to capture temporal dependencies and conduct dimensionality reduction, particularly well-suited for handling sequence data, which is prevalent in domains like time-series analysis, signal processing, and natural language processing.

The hybrid model comprises two seamlessly integrated components: the LSTM layers and the Autoencoder framework. Initially, the LSTM layers serve as the encoder, proficiently capturing the temporal dynamics by processing the input sequence data through their recurrent connections and gated mechanisms that control information flow. The LSTM's unique ability to selectively remember and forget information allows it to encode the sequence into a fixed-sized vector at its bottleneck, encapsulating the temporal characteristics of the input sequence.

Subsequently, this vector serves as the encoded representation, which is then inputted into the Autoencoder segment of the architecture. The Autoencoder typically features a decoder that mirrors the encoder's architecture but in reverse. Its goal is to reconstruct the original sequence from the LSTM's encoded vector. The reconstruction phase is pivotal as it compels the network to learn a compressed representation of the sequence data that retains its essential temporal characteristics.

The training process of a hybrid LSTM-Autoencoder involves backpropagation through time (BPTT), during which the model learns to minimize the dissimilarity between the original input sequence and the reconstructed sequence output by the decoder. This iterative process encourages the LSTM to capture significant temporal features while allowing the Autoencoder to enhance its reconstruction abilities. The outcome is a compressed yet informative representation of sequential data.

The elegant fusion of LSTMs and Autoencoders in this hybrid model provides a powerful tool for feature extraction in sequential datasets. It has proven particularly valuable in anomaly detection within time-series data, where the model is trained on normal sequences, and anomalies are identified as deviations from the learned normal patterns during the reconstruction phase. Moreover, in natural language processing, such a model can distil the essence of sentences or paragraphs into a lower-dimensional space, offering benefits for tasks like text summarization or machine translation, where contextual and temporal aspects of language play a crucial role.

The hybrid LSTM-Autoencoder model represents a substantial advancement in our capacity to distil and interpret complex sequential information, offering an elegant solution to some of the most formidable challenges in the field of DL.

CNN-LSTM Hybrid

This CNN-LSTM hybrid approach harnesses the respective strengths of both architectures to address intricate challenges commonly encountered in domains such as video classification, NLP, and time series analysis. These fields often involve data with significant spatial and temporal dynamics, making integrating spatial and sequential information crucial.

With a focus on acquiring spatial hierarchies from the input data, the CNN component in this hybrid architecture functions as a hierarchical feature extractor. A series of convolutional layers and pooling layers is typically used to minimise spatial dimensions and retain critical information. This stage makes extracting spatial characteristics easier and reduces dimensionality by using techniques like pooling, convolution, and non-linear activation. The output is an abstraction layer representing the input data at several levels.

The CNN extracts spatial features, and then the LSTM component steps in to represent the temporal and sequential dependencies in the data. Specialized RNN called LSTM units, have the remarkable capacity to learn and remember patterns across long sequences. Because they incorporate gate structures—the input gate, forget gate, and output gate—they can successfully solve the disappearing or exploding gradient issues inherent in regular RNNs. Together, these gates control the information flow by determining what needs to be remembered and what needs to be discarded at each time step of the sequence. More precisely, the LSTM network receives the spatial information that the CNN derived as a series of inputs. The LSTM layers must determine the temporal connections between these sequential inputs to understand patterns that change over time. The LSTM network can produce a series of outputs or a final prediction based on the obtained sequential representations. Two essential steps in training a CNN-LSTM hybrid model are regular backpropagation for the CNN segment and backpropagation via time for the LSTM segment. This procedure minimizes a predetermined loss function, which measures the difference between the predicted and actual target values, by optimizing the model parameters using gradient descent or its variations.

A crucial element in efficiently utilizing the CNN-LSTM hybrid architecture is the deliberate arrangement of the parameters and structure of both the CNN and LSTM constituents. Together, the two work more effectively in this form, which combines temporally astute analysis from the LSTM network with spatially rich representations from CNN. The end outcome is a thorough grasp of the temporal and spatial dynamics present in the data.

The use of the CNN-LSTM hybrid architecture has been essential in increasing cutting-edge performance in several different fields. This includes fields such as video analysis, where allows temporal dynamics to be understood via LSTM and spatial information to be

understood simultaneously using CNN. Through carefully examining sequential data, the architecture excels in capturing the structural complexities and connections found in textual data for natural language processing. The CNN-LSTM hybrid model is an effective tool for managing complex data types with rich spatial and temporal dynamics because of its adaptability and resilience.

Our method makes use of the CNN-LSTM architecture's advantages. Our architecture starts with an embedding layer that converts each word in the input sequence into a dense vector representation, much to the CNN model. The model then uses one-dimensional convolutional layers (Conv1D), with a kernel size of three and filters of 64 and 128. A max-pooling procedure is implemented after every convolutional layer to minimize computational cost through the reduction of feature map dimensionality.

After the max-pooling and convolutional layers, a crucial 64-unit LSTM layer is added. This layer makes the ability to capture temporal dependencies within the data sequences possible. We include a dropout layer with a rate of 0.2 to reduce overfitting. In the end, we use a 64-unit dense layer with ReLU activation to complete the architecture and then a second, 2-unit dense layer with sigmoid function activation that is very useful for binary classification applications.

CNN-GRU Hybrid

This advanced hybrid model is well-suited for DL applications that demand a smooth interaction between temporal and spatial dynamics since it is carefully designed to perform well in extracting spatial patterns and capturing sequential information [89].

An increasing filter hierarchy designates each convolutional layer comprising this algorithm's first phase. It starts with 64 filters and moves on to a layer that is 128 times larger. From the input data, these convolutional components precisely extract hierarchical features. The addition of max-pooling layers following each convolutional phase highlights this process. These layers employ a mechanism described in the cited literature to minimize spatial dimensionality while keeping important properties deliberately.

Moving on to the architecture's sequential component, a crucial 64-unit GRU layer is activated. This GRU layer is the system's central component, coordinating the careful examination of temporal dependencies with its unique gating techniques. At each stage of the sequence, these processes control the flow of information by deciding which data should be kept or destroyed. This improves the model's memory efficiency and qualifies it for jobs requiring a sophisticated comprehension of historical contexts or occurrences.

To strengthen the resilience of the architecture, a dropout layer is inserted at a deliberate rate of 0.2. By lowering the chance of adjusting the model too closely to noise in the training data, this method attempts to prevent overfitting and improve the model's generalisation capacity.

The ReLU function activates 64 units in a dense layer that is part of the architectural hierarchy's subsequent layers. The architectural design results in a dense output layer with two units driven by a sigmoid function. This mathematical architecture may generate probabilistic results and can accurately and discernly provide class-specific outputs.

The CNN-GRU hybrid model is a comprehensive solution that neatly combines temporal and spatial analyses into a unified framework. Its synergistic integration of recurrent and CNN paradigms positions it to extract intricate patterns from data. This model offers a potentially useful way to improve prediction accuracy for current DL projects.

3.2.5. Evaluation measures

In our study, the evaluation of the implemented classifiers plays a pivotal role in comprehending the effectiveness of each model in fulfilling its designated task [124]. To achieve this understanding, we employ a set of performance metrics that offer diverse perspectives on the classifiers' efficacy. These metrics are rigorously defined through mathematical equations.

Recall (REC) measures how well the model can locate all relevant instances in the dataset. Higher REC values indicate that the classifier misses fewer real positive cases.

$$REC = \frac{TP}{FN + TP}$$

Precision (PREC) assesses how well the model predicts favourable outcomes. A high accuracy number means that the model makes fewer erroneous optimistic predictions.

$$PREC = \frac{TP}{TP + FP}$$

ACC is an easily comprehended indicator that provides a broad assessment of the model's overall performance. The ratio of accurately identified instances to the total number of occurrences in the dataset, including false positives and FN, is used to calculate it.

$$ACC = \frac{TP}{TP + TN + TN + FP + FN}$$

Precision and REC are considered in equal measure by the F1-Score (F1). Calculating the harmonic mean of these two metrics provides a fair evaluation of the model's effectiveness, particularly in situations where there are class disparities or if both FP and FN have important consequences.

$$F1 = \frac{2 \times REC \times PREC}{REC + PREC}$$

To provide clarity, let's define the terms associated with these metrics:

- True Positives (TP): These are the situations in which the model accurately classifies a news story as positive, indicating that the classification the model predicted and the actual classification agree.
- True Negatives (TN): These are situations where the model correctly labels a news item as negative, showing coherence between the expected and real categories.
- False Positives (FP): These situations arise when a negative news piece is mistakenly classified as positive by the model, leading to a prediction that is at odds with the real category.
- False Negatives (FN): These occur when a positive news piece is inadvertently classified as a negative by the model, indicating a difference between the expected and actual categories.

While each of these indicators provides a unique viewpoint for assessing classifier performance, taken as a whole, they offer thorough insights that guide both model selection and future research areas.

3.3. Conclusion

This chapter carefully outlines our approach to address the FND difficulty in our study. Our methodology covers several important phases, from gathering data and preprocessing to putting the model into practice, training it, and assessing it. These stages collectively form a structured and comprehensive approach for conducting a thorough empirical investigation into this intricate problem.



4. EXPERIMENTAL RESULTS

4.1. Introduction

In this chapter, we thoroughly examine the performance metrics of several ML and DL models on several datasets focused on FND. Our study includes both DL techniques and a range of conventional ML classifiers. By using a variety of datasets, including ISOT, FakeNewsNet, Dataset1, and Dataset2, we can assess how these models' efficacy changes with the type and complexity of the data. Our goal is to offer a detailed understanding of the advantages and disadvantages of each model by looking at important performance metrics. The results highlight the important role that strategic model selection plays in establishing successful FND by illuminating significant variations in model performance.

4.2. The Training Parameters

We used DL models in our research and carefully adjusted their hyperparameters to achieve the best possible results. In particular, our preference for classification problems containing numerous classes led us to select the categorical cross-entropy loss function. We utilized the well-respected Adam optimizer for the critical optimization task because of its reputation for efficiency and efficacy in deep neural network (DNN) training.

We ran the training throughout five epochs to give an overview of the procedure. An epoch in this context denotes a full cycle of all training instances passing through the model in both forward and backward directions. With great care, the batch size was chosen at 128. This means that after processing 128 data, the optimization algorithm updated the model's weights. This batch size balances the granularity of weight updates and computational performance.

The architecture of our models was thoughtfully created to accommodate many circumstances, enabling categorization into two, three, or four unique groups. To fully assess the performance of these models, we employed k-fold cross-validation, with k set to 5. Using this technique, the dataset is split into five equal-sized "folds," or subsets. For each validation iteration, one fold serves as the validation set, and the remaining four are the training set. All five cross-validation cycles include training the model over many epochs. After each

epoch, important metrics, such as loss and ACC, are computed for the training and validation sets. This continuous evaluation of the model's performance provides valuable information about its generalizability, especially its ability to produce accurate predictions on new, untested data.

4.3. Results of Binary Classification Using the ISOT Dataset

4.3.1. Machine learning classifiers results

Random Forest:

In scrutinizing the CM in Figure 4.1 derived from the RF model tailored for FND, as depicted, a discernible proficiency in distinguishing between fake and real news articles is evident. The matrix reveals that out of the articles labelled as fake, 470 were accurately identified as such, reflecting the TN, whereas 290 instances were incorrectly classified as real, indicating the FP. On the flip side, for the articles marked initially as real, 3,198 were rightly classified, signifying TP, and a relatively minor number of 681 cases were mistakenly labelled as fake, pointing to the FN.

This matrix indicates a commendable TP rate of roughly 92.3%, calculated from the 3,198 correctly identified real news instances out of the 3,879 total real news cases (3,198 TP + 681 FN). In addition, the PREC for the classification of real newsstands at 91.7%, deduced from the 3,198 TP identifications from the collective count of 3,488 predictions made as real (3,198 TP + 290 FP). These metrics underscore the model's robustness, especially its high sensitivity in accurately recognizing real news, a critical aspect of information validation.

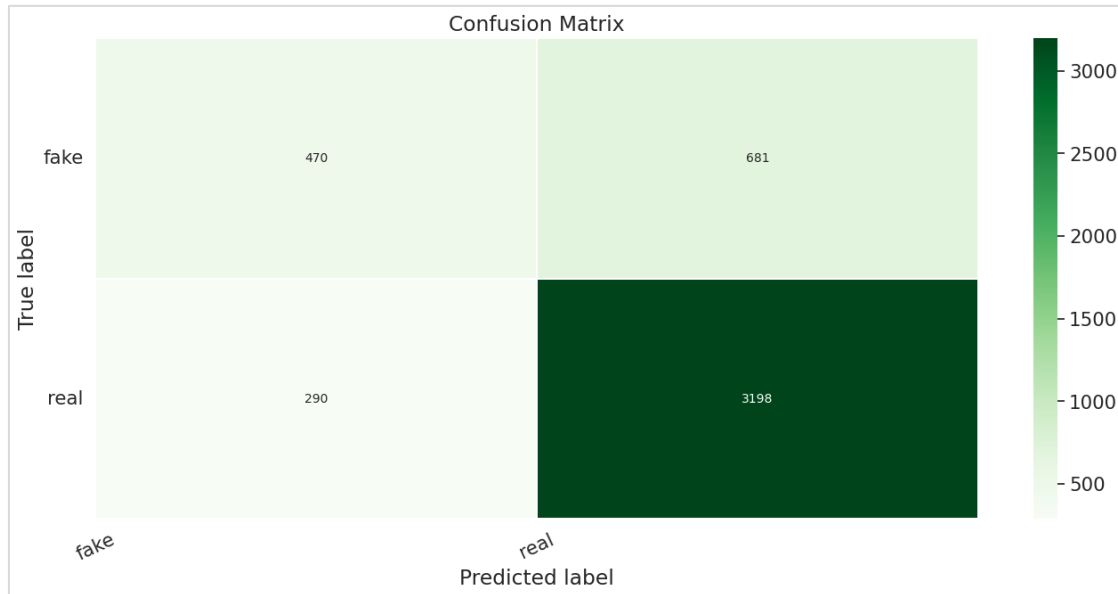


Figure 4.1. Evaluation Matrix for RF Classifier

In evaluating the efficacy of the RF classifier for FND, the CR provides nuanced insights into the model's performance. From a dataset comprising 4,639 instances, the model has attained an overall ACC of 79%, which reflects a competent level of PREC in discerning the authenticity of news articles.

Delving deeper into the classification metrics, the PREC for the negative class (class 0, presumably 'fake') is recorded at 0.62, and for the positive class (class 1, presumably 'real') at 0.82. This PREC metric indicates that the model predicts real news articles more accurately than fake ones. The REC, which assesses the model's ability to identify all relevant instances, is 0.41 for fake news and 0.92 for real news. This suggests that the model is significantly better at detecting real news as it captures most TP instances.

The F1-S, which harmonizes the PREC and REC, is 0.49 for fake news and a much higher 0.87 for real news. This discrepancy suggests that while the model is robust in identifying real news (class 1), its performance in identifying fake news (class 0) is not as strong and could benefit further refinement.

When examining the macro averages, which consider both classes equally, we find a REC of 0.66 and an F1-S of 0.68, hinting at moderate performance across both classes. However, the weighted averages—which account for the support, the number of true instances for each class—

show an F1-S of 0.77, indicating a better performance due to the model's stronger results on the more heavily represented class.

Lastly, we can infer an error rate of about 21% from the given accuracy, signifying the proportion of misclassified instances. Although the accuracy is strong, the error rate highlights the potential for improvement, particularly in the PREC and REC for the fake news category.

Classification Report:				
	precision	recall	f1-score	support
0	0.62	0.41	0.49	1151
1	0.82	0.92	0.87	3488
accuracy			0.79	4639
macro avg	0.72	0.66	0.68	4639
weighted avg	0.77	0.79	0.77	4639

Figure 4.2. Performance Evaluation of RF Classifier

Gradient Boosting

The GB classifier, a potent EL technique, was employed in our binary classification task to discern between fake and true news. Across a five-fold cross-validation framework, the algorithm's performance was rigorously assessed to determine its efficacy.

Beginning with fold one, the model attained an initial test ACC of 81.68%, a robust start to the classification process. Fold two increased, reaching an ACC of 82.32%, suggesting the model's ability to effectively adapt and learn from the training data. The third fold showed a further incremental improvement, achieving an ACC of 82.69%, the highest across all folds, indicating the model's potential for accurately distinguishing fake news when conditions are optimal. However, in folds four and five, there was a slight decrease in ACC, registering at 81.96%. This consistent result across the final two folds suggests a stabilization in the model's performance, although slightly lower than the peak in fold three. The overall test ACC, as gauged by the ACC_score function following cross-validation, was confirmed to be 81.96%.

The CM for the GB model used in FND reveals a dichotomy in classification performance between real and fake news (Figure 4.3). The model displays a commendable REC rate when considering the real news category, correctly classifying 3,486 out of 3,488 instances as real.

This translates to an almost perfect REC rate of 99.94%, underscoring the model's exceptional ability to identify real news.

On the other hand, the fake news category presents a challenge. Of the instances labelled as fake, only 316 are accurately identified by the model, constituting TN. A significant number, 835, are incorrectly classified as real, denoting FP. This indicates a specificity of 27.47%, computed by the ratio of TN to the sum of TN and FP. This lower specificity points to a critical area for improvement in the model's capacity to discern fake news accurately.

The overall picture painted by this CM is one where the GB model is highly adept at recognizing real news but struggles with correctly identifying fake news, necessitating further refinement of the model to enhance its discriminative power across both categories.

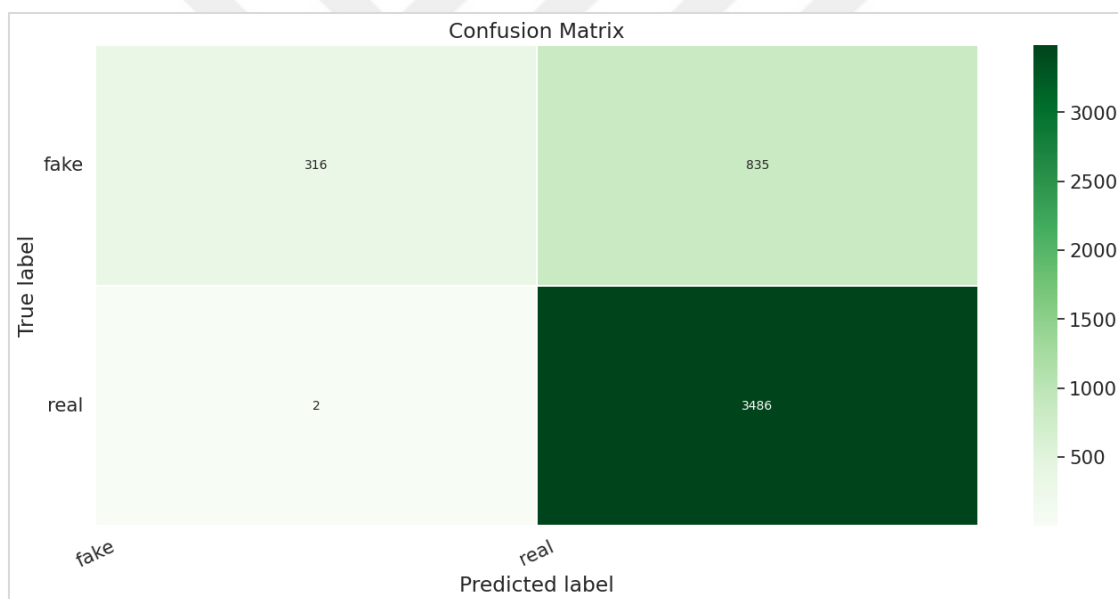


Figure 4.3. Evaluation Matrix for GB

Upon evaluating the CR for the GB model applied to FND, we notice a marked contrast in performance between the classes representing fake (0) and real (1) news. The model exhibits a commendable overall ACC of 82%, indicating that most news articles were correctly classified.

For class 0 (fake news), the PREC is outstanding at 0.99, meaning that virtually all articles predicted as fake were fake. However, the REC for fake news is quite low at 0.27, suggesting that the model only identified 27% of all fake articles, missing many fake news instances. This

discrepancy is reflected in an F1-S of 0.43, indicating that while the model's predictions of fake news are very accurate, it frequently fails to label fake news correctly.

In contrast, for class 1 (real news), the model achieves perfect REC at 1.00, correctly identifying every real news instance. The PREC for real news is 0.81, indicating that among the articles classified as real, 19% were not real. This results in a high F1-S for real news at 0.89, showing a strong balance between PREC and REC for this class despite the presence of FP.

Considering both classes equally, the macro averages yield a PREC of 0.90 and an F1-S of 0.66, pointing towards a solid overall predictive performance with potential improvement areas, particularly in balancing out the performance across classes. The weighted averages, skewed by the larger support for class 1, present an F1-S of 0.78, reflecting a decent performance when considering the class distribution yet indicating that the model's performance on the minority class (fake news) could be improved for better balance.

Classification Report:				
	precision	recall	f1-score	support
0	0.99	0.27	0.43	1151
1	0.81	1.00	0.89	3488
accuracy			0.82	4639
macro avg	0.90	0.64	0.66	4639
weighted avg	0.85	0.82	0.78	4639

Figure 4.4. Performance Evaluation of GB Classifier

Logistic regression

In evaluating fake news classification using LR, a commonly used statistical method for binary outcomes, the model underwent a five-fold cross-validation to ascertain its performance. The results indicate a relatively consistent accuracy across all folds, hovering around the 75%. Precisely, fold one commenced with an ACC of 75.19%. Slight variations were observed in the subsequent folds, with fold two recording an ACC of 75.06%, fold three with 75.19%, fold four at 75.12%, and fold five showing a marginal increase to 75.21%.

The overall test ACC, calculated after the cross-validation, aligns with the highest fold ACC at 75.21%. These figures suggest that while LR provides stable performance in classifying news as fake or real, there is a significant proportion of instances—nearly one quarter—

where the model's predictions are not aligned with the actual labels. This level of ACC, though reasonable, leaves room for improvement in enhancing the model's predictive capabilities.

Analyzing the CM from the LR model utilized for FND, we note significant discrepancies in the classification ACC for fake and real news. The model demonstrates a high proficiency in identifying real news, with 3,486 out of 3,488 actual real news instances correctly classified, resulting in an exceptionally high sensitivity rate. This translates to a near-perfect TP rate, with only 2 instances of real news mistakenly classified as fake, underscoring the model's PREC.

In stark contrast, the model's performance in identifying fake news is notably poor. Of the 1,151 actual fake news instances, only 3 were correctly identified, yielding a TN rate that is alarmingly low. The vast majority of fake news instances, 1,148, were incorrectly classified as real. This result indicates a significant issue with the model's ability to discern fake news, as reflected by a high false positive rate.

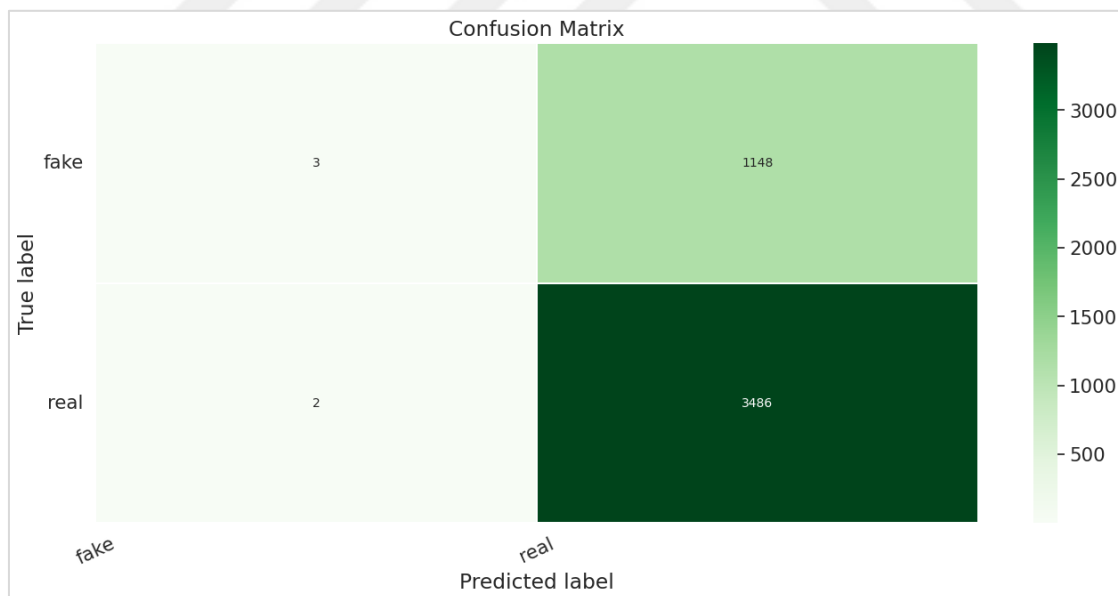


Figure 4.5. Evaluation Matrix for LR

The CR for the LR model in the context of FND reveals a mixed performance. With an overall ACC of 75%, the model correctly classified three-quarters of the cases in the test set. However, the class-specific performance metrics highlight a significant discrepancy between the two classes.

The PREC for class 0, or bogus news, is low at 0.60, meaning that only 60% of the cases that are labelled as fake are, in fact, fraudulent. A significant flaw in the model is that it could not accurately detect any examples of bogus news (REC for this class is 0.00). The F1-S further supports this for class 0, which is at 0.01 and indicates that the model is essentially useless for identifying false news.

Conversely, the model performs much better with class 1, representing real news, with a REC of 1.00, correctly identifying all real news instances. The PREC for class 1 is 0.75, suggesting that while the model is reliable in identifying real news, there is still a 25% rate of FP within this category. The F1-S for class 1 is quite high at 0.86, demonstrating a strong balance between PREC and REC for the real news classification.

When considering the macro average, we see a PREC of 0.68 and an F1-S of 0.43, which suggests that the overall effectiveness of the model is moderate. The weighted average F1-S, which accounts for the support or number of instances in each class, stands at 0.65, indicating that when class distribution is considered, the model's performance is somewhat better, but still not without significant room for improvement.

The error rate, implicitly reflected in the ACC, indicates that about 25% of the predictions made by the model are incorrect, affirming that while the LR model can classify real news with high ACC, its performance in detecting fake news is severely lacking and requires considerable enhancement.

Classification Report:				
	precision	recall	f1-score	support
0	0.60	0.00	0.01	1151
1	0.75	1.00	0.86	3488
accuracy			0.75	4639
macro avg	0.68	0.50	0.43	4639
weighted avg	0.71	0.75	0.65	4639

Figure 4.6. Performance Evaluation of LR Classifier

XGBoost

The XGBoost model, an advanced GB framework, was tested in classifying news articles into 'fake' and 'true'. The evaluation conducted through a five-fold cross-validation revealed a spectrum of accuracies indicative of the model's generalizability and performance consistency.

The initial fold showed an ACC of 84.40%, a promising start for the classification task. The second fold reported a marginal increase to 84.59%, while the third fold further improved the model's performance, recording the highest ACC of the series at 85.38%. This suggests that the model can adapt and learn from the dataset effectively.

However, a dip in ACC was observed in the fourth and fifth folds, registering accuracies close to 83.60%. This decrease in the latter folds could point towards certain limitations of the model when dealing with different distributions of data or particular features inherent in the subsets of the dataset.

The overall test ACC, calculated after cross-validation, stood at 83.57%.

Upon inspecting the CM in Figure 4.7 from the XGBoost model applied to FND, we note a nuanced performance in classifying fake and real news. The model accurately classified 533 instances as fake for the fake news class. However, 618 instances of fake news were incorrectly labelled as real, highlighting an area for improvement in the model's ability to discern fake news accurately.

In the case of real news classification, the model shows a strong ability to identify real news, with 3,344 instances accurately labelled correctly. Nonetheless, there are 144 instances where real news has been misclassified as fake. The TP rate or REC for real news is very high, at 95.87%, calculated as $3,344/(3,344+144)$. In contrast, the REC for the fake news class is significantly lower, at around 46.21%, computed as $533/(533+618)$, suggesting that less than half of the fake news instances were correctly identified.

The model's overall ACC, which can be derived from the sum of TP and TN divided by the total number of instances, stands at 83.58%, calculated as $(533+3,344)/(533+618+144+3,344)$. This indicates that while the model is quite adept at

identifying real news, its performance in accurately classifying fake news is less reliable and warrants further attention to enhance its predictive ACC.

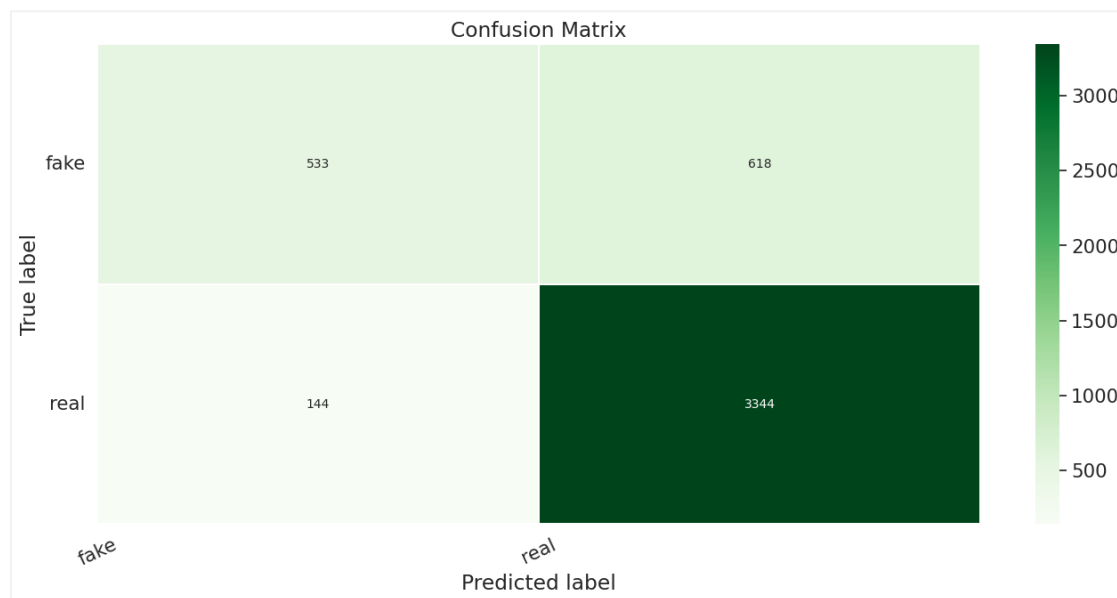


Figure 4.7. Evaluation Matrix for XGBoost

The CR for the XGBoost model utilized in FND highlights a disparity in the model's ability to discern between fake and real news. The model achieves an overall ACC of 84%, indicating a strong ability to classify the majority of the dataset correctly.

Class 0, corresponding to fake news, shows a PREC of 0.79, meaning that 79% of the articles identified as fake by the model were indeed fake. However, the REC for fake news is relatively low at 0.46, indicating that the model only correctly identified 46% of all fake news articles. This results in a moderate F1-S of 0.58, suggesting room for improvement, particularly regarding the model's sensitivity to fake news.

Conversely, the model demonstrates a much stronger performance for class 1, which denotes real news, with a high REC of 0.96. This implies the model successfully identified 96% of the real news articles. The PREC for real news is also relatively high at 0.84, and the resulting F1-S for this class is 0.90, indicating an excellent balance between PREC and REC.

The macro average, which provides an unweighted mean of the performance metrics for both classes, yields an F1-S of 0.74, while the weighted average F1-S is 0.82, accounting for the class distribution.

Classification Report:					
	precision	recall	f1-score	support	
0	0.79	0.46	0.58	1151	
1	0.84	0.96	0.90	3488	
accuracy			0.84	4639	
macro avg	0.82	0.71	0.74	4639	
weighted avg	0.83	0.84	0.82	4639	

Figure 4.8. Performance Evaluation of XGBoost Classifier

Decision Tree

The DT algorithm, a fundamental ML technique, was employed for the binary classification task of discerning 'fake' from 'true' news. Across the cross-validation exercise with five distinct folds, the model showcased a range of accuracies, reflecting its ability to tackle the intricacies of fake news classification.

The model initiated its performance with a test ACC of 84.40% in the first fold, indicating a promising start. Slight improvements were observed in fold two with an ACC of 84.59%, and fold three further elevated this to 85.38%, the peak across the series, suggesting an aptitude for adapting to the dataset's nuances.

A decline was noted in the fourth and fifth folds, with both registering accuracies around 83.60%, pointing to possible challenges the model faces with different data splits or inherent variability within the data. The overall test ACC, capturing the model's generalized performance, was consistent with the final fold at 83.57%.

The CM in Figure 4.9 from the DT model used in FND gives us insight into its classification capabilities. For fake news, out of 1,151 instances, the model correctly identified 533 as fake. However, 618 fake news instances were mistakenly classified as real, highlighting a substantial area for improvement in the model's ability to identify fake news correctly.

Turning to detecting real news, the model performed with greater ACC. It correctly classified 3,344 out of 3,488 instances as real, with only 144 instances of real news incorrectly labelled as fake. This indicates the model's strong ability to recognize real news.

Considering the sum of TP and TN, the model's overall ACC is calculated as $(533 + 3,344) / (533 + 618 + 144 + 3,344)$, which is 84.15%.

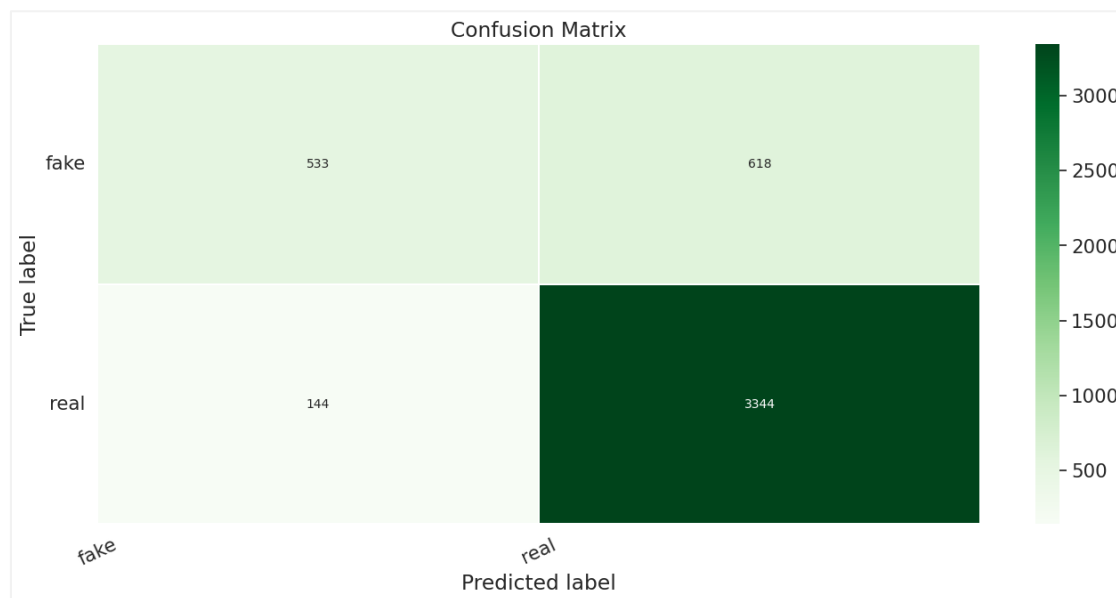


Figure 4.9. Evaluation Matrix for DT

The CR for the DT model in FND illustrates varied effectiveness across the fake and real news classes. With an overall ACC of 84%, the model reliably classified most cases.

For class 0 (fake news), PREC is at 0.79, indicating that 79% of the news articles the model classified as fake were indeed fake. However, the REC for this class is lower at 0.46, meaning that the model only correctly identified 46% of all actual fake news articles. The resulting F1-S for class 0 is 0.58, pointing towards a moderate balance between PREC and REC and suggesting an area for more accurate improvement in identifying fake news.

Conversely, the model shows a strong performance in identifying real news (class 1), with a REC of 0.96, indicating it correctly identified 96% of the real news articles. The PREC for class 1 stands at 0.84, and the F1-S is high at 0.90, denoting an excellent balance between PREC and REC for real news classification.

The macro average across both classes gives us a PREC of 0.82 and an F1-S of 0.74, indicating an excellent overall performance. The weighted average, which considers the support (number of instances in each class), presents an F1-S of 0.82, suggesting that the model's performance is robust when adjusted for class prevalence.

Classification Report:				
	precision	recall	f1-score	support
0	0.79	0.46	0.58	1151
1	0.84	0.96	0.90	3488
accuracy			0.84	4639
macro avg	0.82	0.71	0.74	4639
weighted avg	0.83	0.84	0.82	4639

Figure 4.10. Performance Evaluation of DT Classifier

Multinomial naive Bayes

The MNB model, commonly utilized for text classification due to its simplicity and effectiveness with discrete features, was applied to classifying news as 'fake' or 'true'. The five-fold cross-validation results reveal a model that struggles with this dataset, exhibiting considerable variance in ACC across different folds.

Fold one commenced with an ACC of just over 50%, a baseline suggesting the model could only marginally distinguish between the classes. Fold two experienced a decline in performance, dropping to 48.67%, further suggesting difficulty in classification. A slight increase in fold three to around 50.10% indicates some recovery yet still reflects a model that does not confidently classify the news articles.

The fourth fold presented the lowest ACC of the sequence at 47.10%, raising concerns about the model's consistency and susceptibility to variations within the data. A rebound in fold five to 50.05% ACC did little to alleviate these concerns, suggesting that while the model can occasionally achieve a moderate level of ACC, it is not reliably discerning fake from real news.

The MNB model appears to have limited efficacy in this context, with an average ACC hovering around the 50% mark across all folds.

The CM for the MNB model in the context of FND offers a clear view of its classification performance (Figure 4.11). For fake news (class 0), the model correctly identifies 665 instances as fake, but it also incorrectly labels 486 instances as real, revealing a significant

number of FP. This misclassification suggests the model has room for improvement in accurately discerning fake news.

For real news (class 1), the model correctly classifies 3,344 articles as real, demonstrating a strong capability in identifying legitimate news content. However, there are 144 instances where real news is incorrectly classified as fake, indicating some FN.

The model's performance indicates a stronger ability to identify real news than fake news, with a higher rate of FP in the fake news category. Adjustments to the model's parameters or reevaluating the features used for classification could potentially enhance its ability to identify fake news more accurately. The high number of FP in classifying fake news suggests that the model may be biased towards predicting news as real, a common challenge in unbalanced datasets where one class predominates.

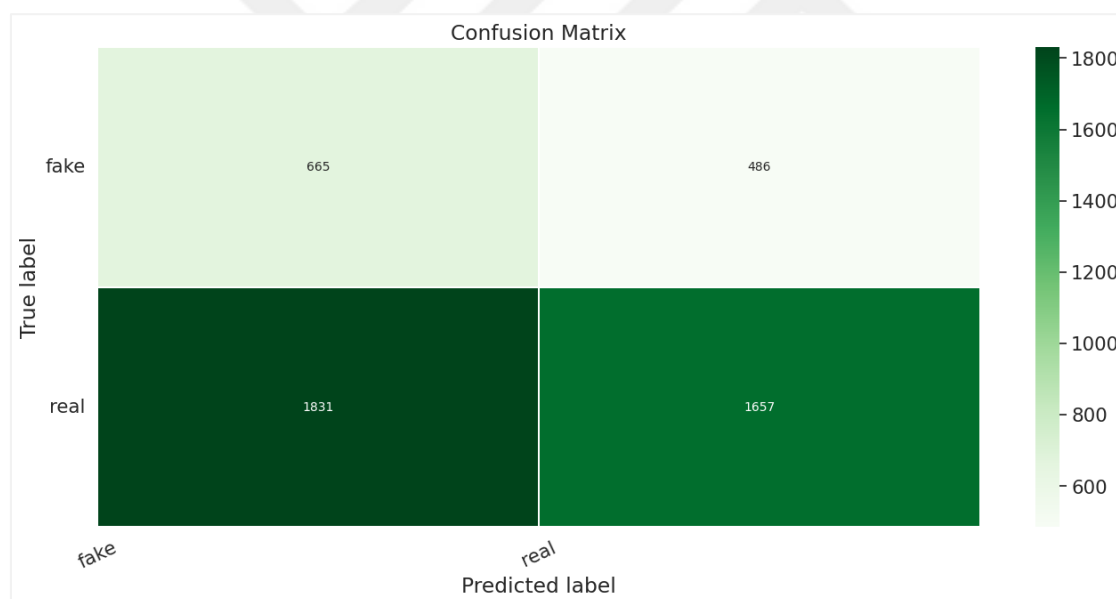


Figure 4.11. Evaluation Matrix for MNB

The CR for the MNB model in the FND task reflects a model with distinct areas for improvement. The overall ACC stands at 50%, which implies that the model is correct in its classification half the time.

Focusing on the fake news category (class 0), the model exhibits a low PREC of 0.27, meaning it identifies only 27% of the fake articles as fake. The REC, however, is higher at 0.58, indicating that the model can identify 58% of the fake news instances in the dataset.

The F1-S for fake news is 0.36, suggesting that the balance between PREC and REC is relatively moderate but leans more towards REC.

In contrast, the real news category (class 1) shows a higher PREC of 0.77, indicating that when the model predicts news as real, it is correct 77% of the time. The REC for real news is lower at 0.48, suggesting that the model captures less than half of real news instances. The F1-S for real newsstands at 0.59 is a more balanced performance than that for fake news but still indicates potential for improvement.

The macro average F1-S across both classes is 0.48, and the weighted average, which considers each class's support, is 0.53.

Classification Report:				
	precision	recall	f1-score	support
0	0.27	0.60	0.37	1165
1	0.77	0.46	0.57	3475
accuracy			0.49	4640
macro avg	0.52	0.53	0.47	4640
weighted avg	0.64	0.49	0.52	4640

Figure 4.12. Performance Evaluation of MNB Classifier

SVM (RBF)

The SVM with Radial Basis Function (SVM-RBF) kernel was applied to the binary classification task of identifying fake versus true news articles, and its performance was evaluated through a five-fold cross-validation. The results from each fold were remarkably consistent, demonstrating the model's stability across different subsets of the dataset.

The SVM-RBF achieved a test ACC of 75.19% in the first fold. This level of ACC was nearly identical across the subsequent folds, with folds two through five, each reporting an ACC of 75.19%. The consistent performance across all folds, all closely huddled around the 75% mark, suggests that the model has a stable and reliable predictive capacity when applied to this classification challenge.

The consistent test ACC of 75.19% indicates that while the SVM-RBF model can classify most articles correctly, there is a significant proportion—nearly a quarter of the dataset—where the model does not predict accurately. This level of performance suggests that the model, with its current tuning and the features used, has room for improvement in enhancing its ability to distinguish between fake and true news more accurately.

The CM for the SVM-RBF model in Figure 4.14 applied to classify news articles into 'fake' and 'real' presents an intriguing outcome. Within this binary classification, the model did not correctly identify any 'fake' news articles, as evidenced by the zero count for TP in the 'fake' news category. This results in a REC of 0% for class 0, indicating that all 'fake' news instances were misclassified. Consequently, the PREC and F1-S for the 'fake' news classification are also 0%, highlighting a significant weakness in the model's capability to detect fake news.

In stark contrast, the model performed with absolute PREC in classifying 'real' news, where it correctly identified all 3,488 instances as 'real', achieving a REC of 100%. This perfect performance for the 'real' news class suggests that while the model is exceptionally good at identifying real news, it fails entirely when distinguishing fake news.

The overall ACC of the model, calculated by considering both TP for 'real' news and the total number of instances, stands at 74.84%, which reflects the model's strong bias towards predicting news articles as 'real'. While seemingly high, this ACC figure is misleading given the complete failure to recognize any 'fake' news. The error rate, which accounts for the incorrect predictions, would be 25.16%, but this does not capture the severity of the model's one-sided performance.

Such results indicate that while SVM-RBF can be very effective for certain classification tasks, it may require significant adjustments, such as parameter tuning or feature engineering, to improve its detection of fake news. This CM indicates that the model's current state is unsuitable for a reliable FND system, as it fails to serve the fundamental need to distinguish between 'fake' and 'real' news.

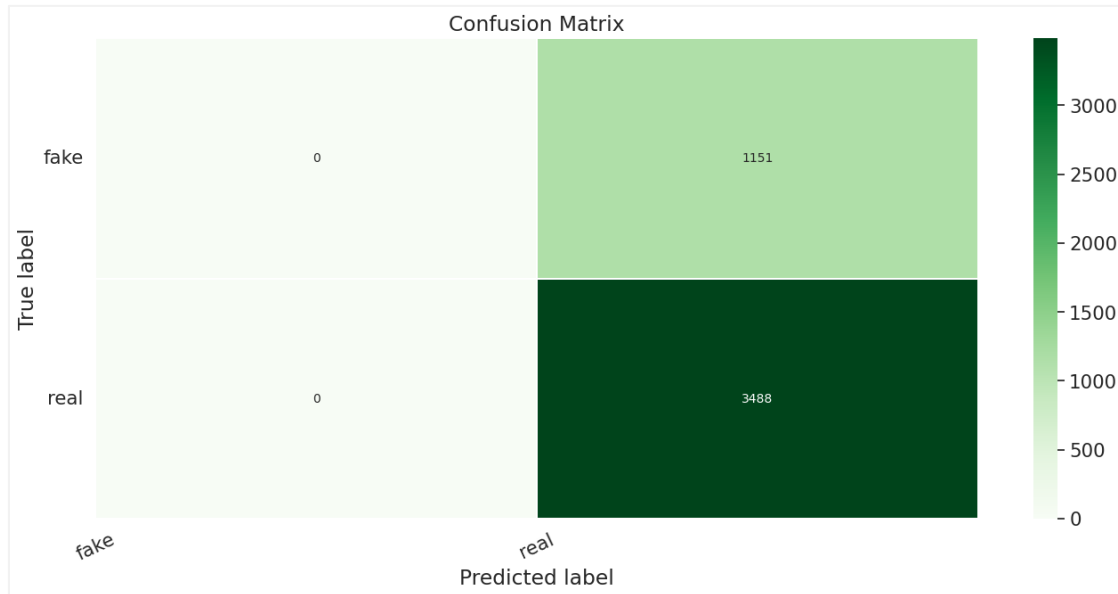


Figure 4.13. Evaluation Matrix for SVM-RBF

The CR for the SVM-RBF model in FND demonstrates a pronounced imbalance in its ability to classify fake versus real news (Figure 4.14) correctly. The model achieves an overall ACC of 75%, indicating that it can correctly classify three-quarters of the dataset.

For fake news (class 0), the PREC and REC are 0.00, indicating that the model fails to identify any fake news articles correctly; every article predicted as fake is incorrect. This is mirrored in an F1-S of 0.00, which suggests the model's ineffectiveness in recognizing fake news within the dataset.

On the other hand, the model exhibits a high PREC of 0.75 for real news (class 1), meaning that when an article is classified as real, it is correct 75% of the time. Additionally, the model has a REC of 1.00 for real news, indicating it correctly identifies all real news articles. This results in a high F1-S of 0.86 for class 1, signifying a strong performance in classifying real news.

The macro averages, which give equal importance to both classes, reveal an average PREC of 0.38 and an average REC of 0.50, leading to a macro F1-S of 0.43. This reflects the significant disparity between the two classes. The weighted average F1-S is 0.65, accounting for the number of instances in each class, indicating a somewhat better performance when considering the class distribution.

The ACC implicitly reflects the error rate, highlighting a significant portion of the predictions being incorrect, primarily due to the model's inability to identify fake news. This suggests a critical need for model improvement or alternative approaches to detect fake news effectively.

Classification Report:				
	precision	recall	f1-score	support
0	0.00	0.00	0.00	1165
1	0.75	1.00	0.86	3475
accuracy			0.75	4640
macro avg	0.37	0.50	0.43	4640
weighted avg	0.56	0.75	0.64	4640

Figure 4.14. Performance Evaluation of SVM-RBF Classifier

Linear SVM

The Linear SVM model's performance in classifying news articles as 'fake' or 'true' over five-fold cross-validation presents a significant variation in ACC, indicating potential instability or sensitivity to the dataset's nuances. As indicated by the warnings across multiple folds, the model's convergence issues suggest that the default iteration limit was insufficient for the model to find an optimal solution, which could have impacted its performance.

In the first fold, the Linear SVM model achieved a test ACC of about 55.34%, a moderate start that suggests some capability in distinguishing between fake and true news. However, the second fold saw a marked drop in ACC to 35.89%, raising concerns about the model's reliability and the need for increased convergence iteration. The third fold showed a slight improvement to 47.90%, but still below a level that would be considered adequate for a classification task of this nature.

Fold four significantly increased test ACC to 60.14%, indicating a better fit between the model and the training data for that particular fold. The final fold saw the highest ACC at 73.21%, suggesting the model has the potential for high ACC under certain conditions.

The CM from the Linear SVM model used in FND suggests that the model has limitations in correctly categorizing fake and real news. For fake news (class 0), the model correctly identified only 47 out of 1,151 instances, with many fake news articles and 1,104 misclassified as real. This

reveals a significant weakness in the model's capability to detect fake news, reflected by the low REC rate (Figure 4.15).

The model performs better for real news (class 1), correctly identifying 3,349 out of 3,488 instances as real. However, it still misclassified 139 real news articles as fake. The high number of TP for real news indicates that the model is more adept at identifying real news than fake news.

The overall ACC of the model, taking into account the sum of TP for both classes, would be calculated as $(47 + 3,349) / (47 + 1,104 + 139 + 3,349)$, which translates to an ACC rate that is significantly better than chance, but not optimal. This points to a strong bias in the model towards classifying news articles as real, potentially due to an imbalance in the dataset or a need for more discriminative feature selection for fake news classification.

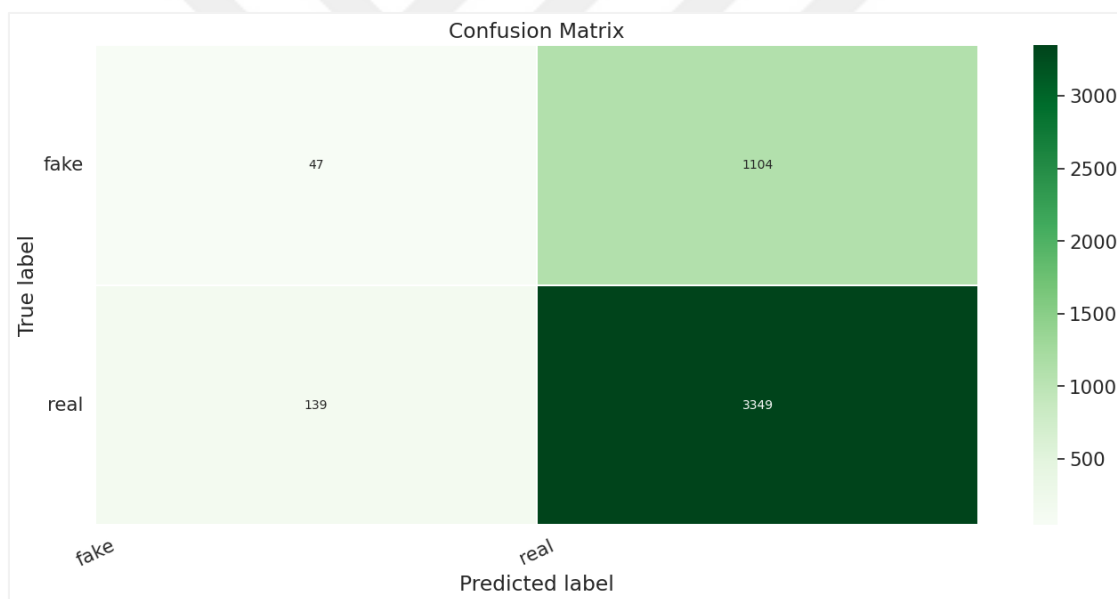


Figure 4.15. Evaluation Matrix for Linear SVM

The CR for the Linear SVM model in the FND task shows a model with a considerable discrepancy in its classification performance for fake and real news. The overall ACC is 73%, suggesting that it correctly classifies most articles (Figure 4.16).

For fake news (class 0), the model has a PREC of 0.25, indicating that only a quarter of the articles predicted as fake are actually fake. The REC for this class is even lower at 0.04, which means the model only correctly identifies 4% of all fake news articles. This results in a low F1-S of 0.07, highlighting the model's struggle to identify fake news accurately.

In contrast, the model performs much better for real news (class 1), with a PREC of 0.75 and a REC of 0.96. This indicates that 75% of the articles classified as real are correct, and the model can identify 96% of all real news articles. The F1-S for real news is high at 0.84, suggesting a strong performance in classifying real news.

The macro averages across both classes show an average PREC and F1-S of 0.50 and 0.46, respectively, which indicates moderate performance. The weighted average, which accounts for the number of instances in each class, shows a better overall F1-S of 0.65.

Classification Report:				
	precision	recall	f1-score	support
0	0.25	0.04	0.07	1151
1	0.75	0.96	0.84	3488
accuracy			0.73	4639
macro avg	0.50	0.50	0.46	4639
weighted avg	0.63	0.73	0.65	4639

Figure 4.16. Performance evaluation of Linear SVM classifier

Comparison

In our investigation, we applied various ML models to classify news articles within the ISOT dataset as either fake or real. The principal measure for assessing the performance of these classifiers was accuracy.

From the data presented in Table 4.1, it is clear that the RF and XGBoost models were among the top performers. However, contrary to the summary provided, the RF model achieved an ACC of 79%, with PREC of 62% and REC of 41% for fake news, and PREC of 82% and REC of 92% for real news, leading to F-S of 49% for fake news and 87% for real news. These figures indicate that while the RF model is proficient in classifying real news, it could be improved for FND.

The XGBoost model reported an ACC of 83.57%, with a PREC of 79% and REC of 46% for the fake news class, while for real news, it had a PREC of 84% and a REC of 96%, resulting in F-Scores of 58% and 90% respectively. These statistics illustrate that XGBoost is particularly effective in identifying real news articles.

Other classifiers displayed varying levels of ACC. GB had an ACC of 81.9%, with a high PREC of 99% but a low REC of 27% for fake news. For real news, it showed a PREC of 81% and a REC of 100%, with F-Scores of 43% for fake news and 89%.

The Linear SVM model had a lower ACC of 73.20%, with significantly lower performance detecting fake news, as evidenced by a PREC of 25%, REC of 0.04%, and an F-Score of 0.07%. For real news, it had a better performance with a PREC of 75% and REC of 96%, resulting in an F-Score of 84%.

The SVM with RBF kernel and LR had moderate ACC, around 75.18% and 74.82%, respectively, but did not correctly identify any fake news instances, as indicated by 0% in PREC and REC for the fake news class.

The DT model showed an ACC of 76.15%, with a relatively balanced performance with a PREC of 52% and REC of 55% for fake news and a PREC of 85% and REC of 83% for real news, leading to F-Scores of 53% and 84% respectively.

Lastly, the MNB had the lowest ACC at 50.53%, and similar to the SVM-RBF and LR, it failed to correctly classify any fake news instances, with 0% PREC and REC for class 0.

Table 4.1. ML classifiers results on ISOT dataset

Classifier	ACC	Class	PREC	REC	F-S
RF	79%	Fake	62%	41%	49%
		Real	82%	92%	87%
GB	81.9%	Fake	99%	27%	43%
		Real	81%	100%	89%
XGBoost	83.57%	Fake	79%	46%	58%
		Real	84%	96%	90%
Linear SVM	73.20%	Fake	25%	0.04%	0.07%
		Real	75%	96%	84%
RBF SVM	75.18%	Fake	0%	0%	0%
		Real	75%	100%	86%
LR	74.82%	Fake	29%	0%	0%
		Real	75%	100%	86%
DT	76.15%	Fake	52%	55%	53%
		Real	85%	83%	84%
MNB	50.53%	Fake	0%	0%	0%
		Real	75%	100%	86%

4.3.2. Deep learning classifiers results

CNN

During the training of our DL model across five distinct folds, a consistent pattern of performance improvement emerged over successive epochs. However, this pattern displayed nuanced variations in both loss and ACC metrics. Each fold consisted of a sequence of 5 epochs involving incremental training and validation steps, as illustrated in Figure 4.17.

In the initial fold, there was a noteworthy reduction in training loss, plummeting from 0.5168 to a minimum of 0.1435, accompanied by a corresponding increase in training ACC from 76.63% to 94.75%. However, during validation, a divergence was observed. Despite an initial decline in loss and an uptick in ACC, the final epochs exhibited an increase in validation loss and a slight dip in validation ACC, resulting in a test ACC of 80.02%.

Subsequent folds revealed a consistent trend of decreasing loss and improving ACC as the epochs progressed. In the second fold, the model achieved significant milestones, culminating in a remarkable test ACC of 92.84%, showcasing a substantial enhancement in predictive performance. The third and fourth folds followed a similar pattern of improvement, with the latter achieving impressive test accuracies of 97.52% and 98.19%, respectively. These results underscore the model's growing proficiency in making accurate predictions.

In the final fold, the validation ACC reached its zenith in the first epoch at 99.16%. However, it experienced a slight decline in the final epochs, ultimately settling at a noteworthy test ACC of 98.60%. The graphical representation of these dynamics, as depicted in the plotted loss and accuracy curves for training and validation phases across the epochs, is compelling evidence of the model's robustness. These visualizations illustrate the trajectories of accuracy and loss metrics in each phase, facilitating a comprehensive analytical understanding of the model's performance throughout different epochs.

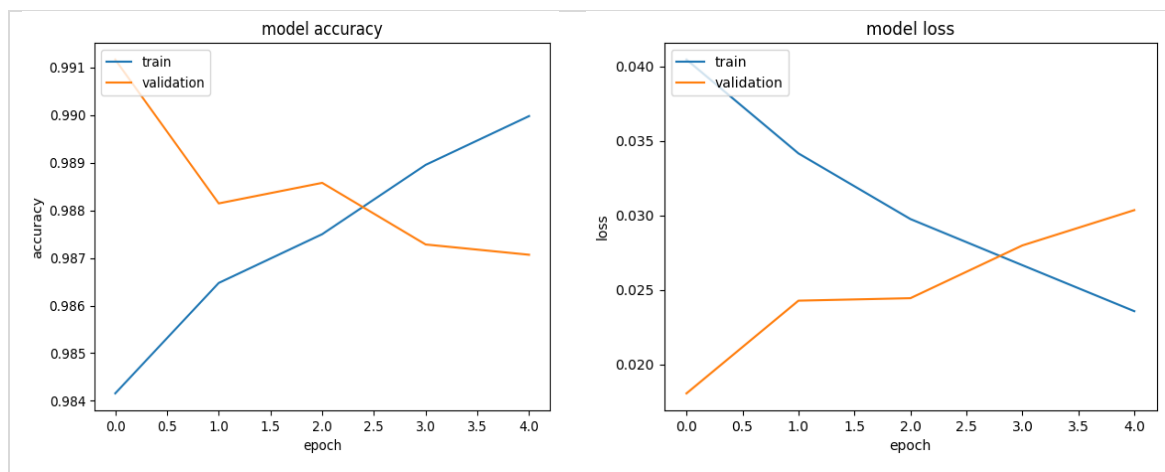


Figure 4.17. Accuracy and loss metrics for training and validation of CNN

In the analysis of the CM generated during the evaluation of the CNN model for FND (Figure 4.18), a notable distinction is observed between the TP and TN, both of which hold significant relevance in the context of misinformation classification.

Among the total predictions, a substantial majority has been accurately classified, with 1134 TN and 3440 TP. This underscores the model's proficiency in correctly discerning fake and real news articles. However, it is essential to address the presence of FN and FP in the matrix, which stands at 48 and 17, respectively.

While these numbers are relatively low, indicating a controlled error rate, they highlight the need for further refinement and optimization of the model. These misclassifications present a significant challenge in FND, albeit a small fraction. They have the potential to facilitate the spread of misinformation, emphasizing the importance of delving deeper into the underlying factors contributing to these erroneous classifications to enhance accuracy.

The presence of 48 FN signifies instances where the model incorrectly identified fake news as real, necessitating meticulous attention to reduce the risk of misinformation dissemination. Similarly, 17 FP represents cases where real news was inaccurately flagged as fake, potentially impeding the circulation of truthful information.

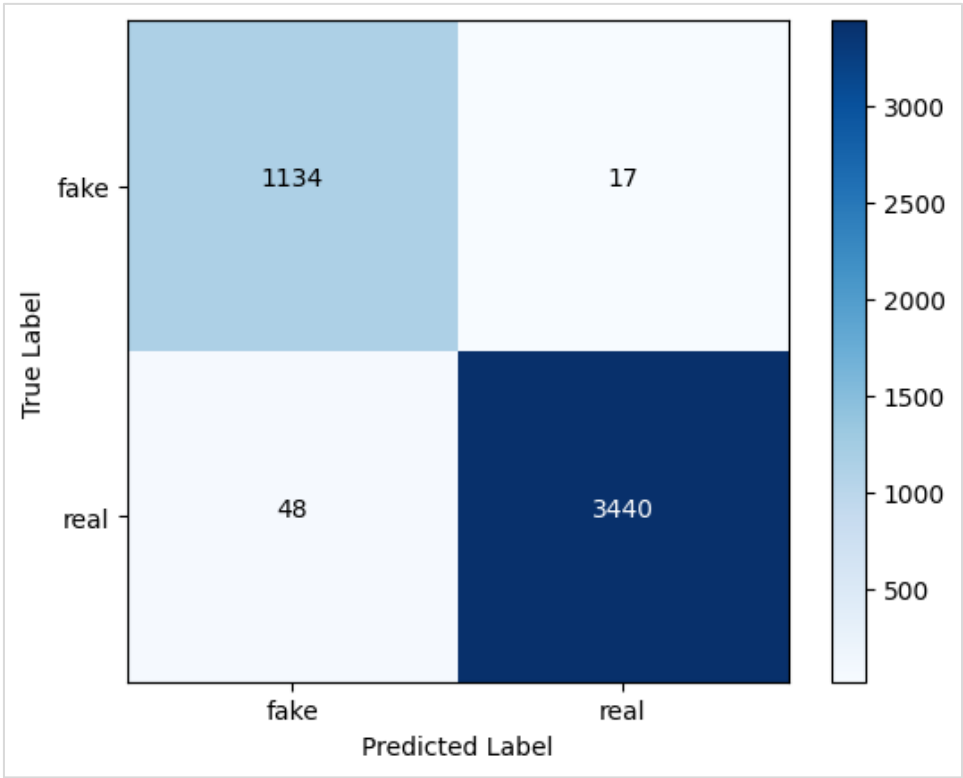


Figure 4.18. Evaluation matrix for CNN

In our most recent application of the CNN model in FND, we have achieved auspicious results, as illustrated in the CR provided (Figure 4.19). The model's ability to distinguish between genuine and fabricated news narratives is notably high, with PREC metrics of 0.96 and 1.00 for categories 0 and 1, respectively. This signifies the model's almost flawless capability to accurately categorize items, particularly with a low false positive rate for category 1.

Our attention then turns to REC values, where category 0 exhibits a REC of 0.99, and category 1 also follows with a REC value of 0.99. These metrics underscore the model's proficiency in identifying TP accurately, resulting in minimal FN rates. The harmonized F1-S, which provides a balanced assessment considering PREC and REC, stands at 0.97 and 0.99 for categories 0 and 1, respectively. This reaffirms the CNN model's reliability and effectiveness in discerning fake news narratives.

Furthermore, the impressive ACC score of 98.6% solidifies the model's robustness, positioning it as a dependable tool in the ongoing fight against misinformation. However, it is essential not to overlook the error rate, which is 1.4%, indicating a small but existing room for further refinement and optimization to approach a foolproof system.

When considering the macro average and weighted average of these metrics, both hovering around the 0.99 mark, it becomes evident that the CNN model exhibits high reliability across various parameters. This marks a significant breakthrough in the field of FND.

Classification Report:				
	precision	recall	f1-score	support
0	0.96	0.99	0.97	1151
1	1.00	0.99	0.99	3488
accuracy			0.99	4639
macro avg	0.98	0.99	0.98	4639
weighted avg	0.99	0.99	0.99	4639

Figure 4.19. Classification report of CNN

CNN-LSTM

Throughout the training of the CNN-LSTM model during a structured five-fold validation process, a noticeable pattern emerges in the validation loss and ACC metrics across all folds, providing insight into the model's performance trajectory. In the initial epochs, the loss metric follows a decreasing trend, indicating the model's improving ability to make correct predictions while simultaneously increasing in ACC (Figure 4.20).

In the first fold's initial epoch, the model achieved a training ACC of 77.82%, which significantly improved to 97.11% by the fifth epoch. However, a concerning rise in validation loss from 0.4082 to 0.6755 was observed, potentially signalling the onset of overfitting. Despite the increasing validation loss, the validation ACC remained relatively stable, oscillating between 81.77% and 80.11%.

The subsequent folds, from the second to the fifth, all started with notably high accuracies in their first epochs, maintaining impressive figures above 93%. Nonetheless, a closer examination reveals a pattern of diminishing returns as the epochs progressed, manifested in rising validation losses and a gradual decline in validation ACC. This suggests the model may gradually lose its generalisation ability as the epochs advance. The second fold displayed the most significant deviation in validation ACC, reaching a peak of 97.89% and decreasing to 96.53% by the final epoch. Notably, the third fold maintained the highest validation ACC at 97.65% despite a noticeable increase in validation loss over epochs.

As we approach the later stages of validation, the trends somewhat stabilize. Folds four and five show a notable increase in validation loss, while the validation ACC exhibits mild fluctuations. This indicates that the model managed to maintain ACC despite the growing complexity reflected in the loss metrics. The fifth fold concluded with a test ACC of 98.04%, representing a balance in the model's predictive capabilities, albeit with an observable rise in validation loss to 0.0518. This suggests room for further optimizations to achieve a more nuanced performance that balances loss and ACC.

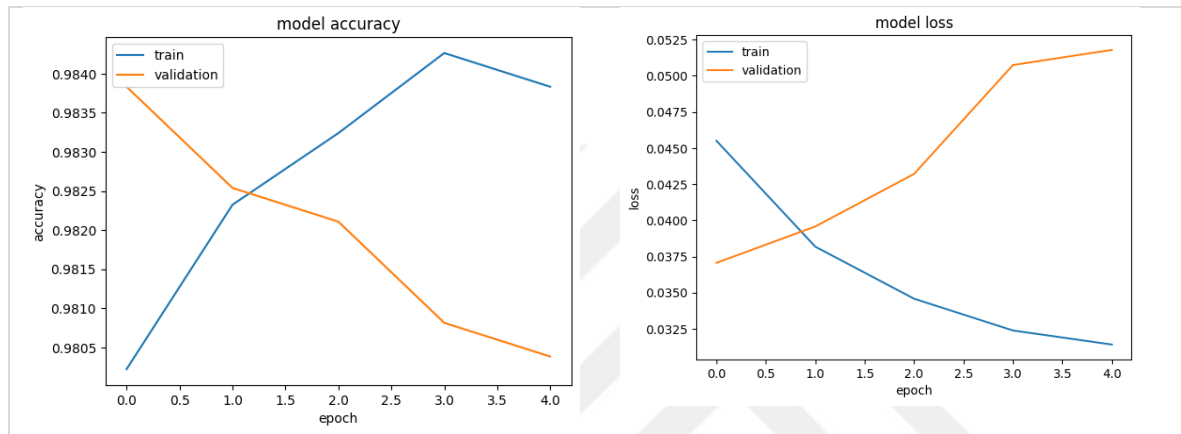


Figure 4.20. Accuracy and loss metrics for training and validation CNN-LSTM

When assessing the CNN-LSTM model's performance in FND, it is crucial to meticulously analyze the data presented in the CM to gain valuable insights into its operational effectiveness. The CM is a pivotal tool for delineating the rates of TP, TN, FP, and FN, providing a snapshot of the model's competence.

In this context, the matrix reveals 3473 TP outcomes, representing instances where fake news was correctly identified, and 1075 TN, denoting the accurate identification of genuine news pieces. Additionally, there are 76 FP, indicating cases where the model incorrectly flagged genuine news as fake, and 15 FN, signifying instances where fake news was mistakenly classified as genuine.

Accordingly, the matrix reveals a noteworthy accuracy (ACC) in identifying fake news, as substantiated by the high true positive (TP) rate. However, the true negative (TN) rate is comparatively lower, suggesting a reduced ability to verify authentic news accurately. This comprehensive analysis of the CM sheds light on the model's strengths and areas for potential improvement in its performance in the challenging task of FND.

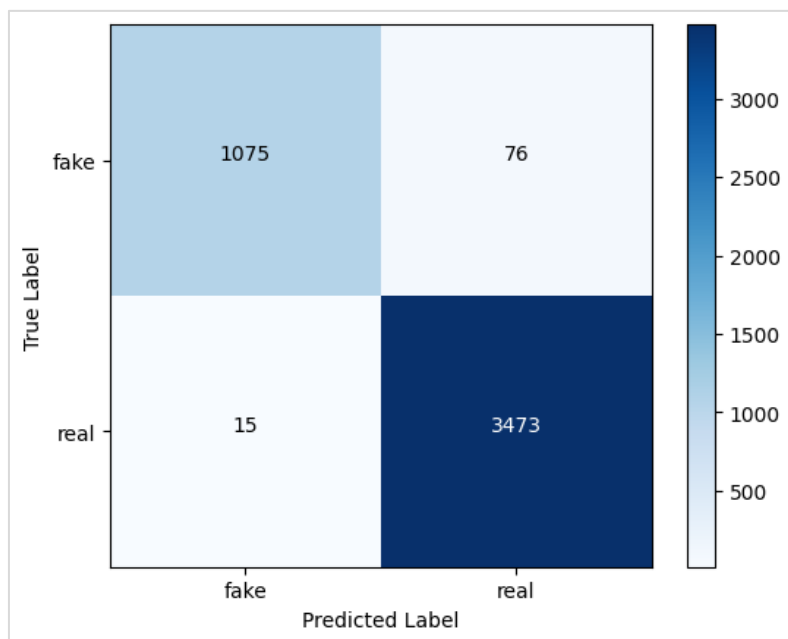


Figure 4.21. Evaluation Matrix for CNN-LSTM

The results derived from the CR assessing the CNN-LSTM model's efficacy in detecting fake news reveal an exceptionally high level of performance. This structured report encompasses a range of statistical metrics, including PREC, REC, F1-S, and support, enabling a meticulous evaluation of the model's proficiency.

For class 0, representing fake news, the PREC impressively stands at 0.99, signifying a 99% success rate in correctly identifying fake news instances among all instances classified as fake news by the model. Furthermore, the REC for this class is 0.93, indicating that the model accurately identified 93% of all actual fake news cases. These metrics yield an F1-S of 0.96, representing a harmonic mean of PREC and REC, offering a balanced perspective of the model's performance in classifying fake news.

Transitioning to class 1, which represents real news, the model reports a PREC of 0.98, denoting a high ACC in correctly distinguishing real news instances. The REC value for this class is an outstanding 1.00, indicating a perfect score in correctly identifying all real news instances. This flawless REC translates to an impressive F1-S of 0.99, affirming exceptional performance in the classification of real news.

From a broader perspective, the model exhibits an overall ACC of 98.04%, calculated as the proportion of correct predictions over the total number of predictions. Correspondingly, the error rate is about 1.96%, representing the proportion of incorrect predictions.

Furthermore, an analysis of the macro average, which computes the mean of scores for both classes, provides a consolidated viewpoint, indicating an excellent balance in classifying real and fake news. This balance is a pivotal aspect in determining the efficacy of the model. The weighted average, which considers the support (number of true instances for each label) in the computation, also reaffirms the high ACC reflected in individual class metrics.

Classification Report:				
	precision	recall	f1-score	support
0	0.99	0.93	0.96	1151
1	0.98	1.00	0.99	3488
accuracy			0.98	4639
macro avg	0.98	0.96	0.97	4639
weighted avg	0.98	0.98	0.98	4639

Figure 4.22. Performance Evaluation of CNN-LSTM Classifier

CNN-GRU

The training and validation process of the CNN-GRU model unfolded over several epochs, aiming to identify and validate its loss and ACC parameters. In the initial fold, a positive progression in training ACC was observed, starting at 77.64% in the first epoch and reaching an impressive 97.25% by the fifth epoch. However, this increase in training ACC came at the cost of validation ACC, which marginally declined from 81.72% to 79.37%. This decline was accompanied by an increase in validation loss, suggesting a potential overfitting scenario where the model excelled in learning the training data but struggled to generalize to unseen data.

The model demonstrated enhanced performance in the second fold, with training ACC starting at a high 93.18% and stabilizing at a high level in subsequent epochs. It maintained notable validation accuracies, consistently above 96%. However, there was a slight increase in validation loss over the epochs, indicating the need for vigilant monitoring to prevent overfitting.

The third fold followed a similar high-performance trend as the second fold, with training ACC starting at 97.60% and showing marginal improvement over the epochs. Validation ACC remained substantially high, hovering around 98%. Nevertheless, there was a slight downward trend in validation ACC as epochs progressed, accompanied by an upward trend in validation loss, suggesting a reduction in the model's ability to generalize well to new data.

In the fourth fold, the model maintained high training ACC, starting at 98.24% and remaining steady in subsequent epochs. However, there was a gradual decrease in validation ACC from 97.84% to 97.00%. This decline was complemented by a steady increase in validation loss, indicating emerging signs of overfitting. Optimization strategies such as regularization or early stopping may be necessary to maintain a healthy balance between training and validation outcomes.

In the fifth fold, the model sustained exemplary performance, with training ACC commencing at 97.95% and showing slight improvement over the epochs. Validation ACC remained strong, hovering around the 98% mark, showcasing commendable generalization performance on the validation set. However, there was a minor increase in validation loss in the last epoch, highlighting the importance of monitoring for potential overfitting tendencies.

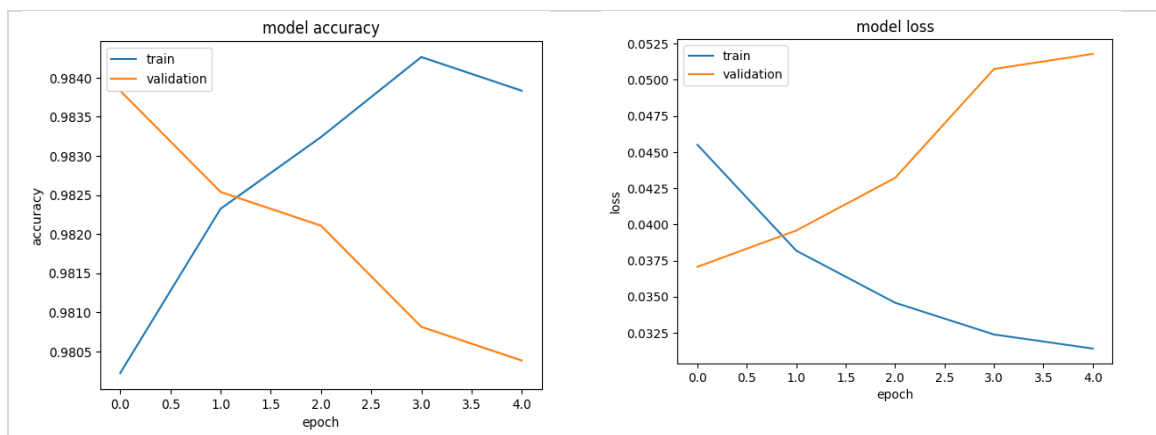


Figure 4.23. Accuracy and loss metrics for training and validation CNN-GRU

In analyzing the outcomes derived from applying the CNN-GRU model for FND, it is essential to examine the presented CM thoroughly. This matrix serves as a valuable tool for assessing the model's performance, categorizing its results into four distinct groups:

Upon scrutinizing the provided matrix, we observe a notable count of 3469 TP, signifying many instances where the model accurately identified news articles as authentic. Additionally, the TN count stands at 1077, indicating that the model proficiently recognized many fake news instances with PREC.

However, it is worth noting that the model exhibited instances of misclassification. Specifically, there were 19 FN, where the model erroneously classified fake news as genuine, and 74 FP, representing cases where the model incorrectly classified genuine news as fake. These misclassifications, while present, constitute a relatively small proportion of the overall dataset, underscoring the need for continuous refinement and optimization further to enhance the model's ACC and reliability in FND.

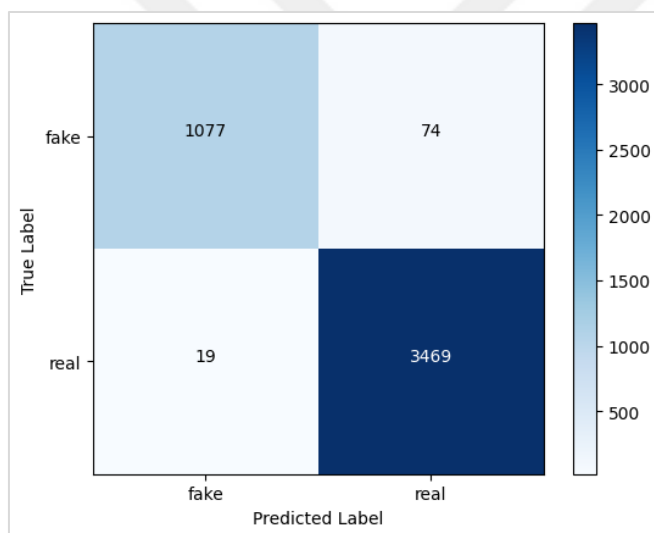


Figure 4.24. Evaluation Matrix for CNN-GRU

Upon evaluating the CNN-GRU model's performance in FND, as elucidated in the CR, we observe a commendable overall ACC of 98.00%. This ACC underscores the model's reliability in distinguishing between genuine and fake news articles. The low % error rate of 2.00% further accentuates the model's efficiency in minimizing incorrect predictions, substantiating its proficiency.

Delving deeper into the performance metrics, we find that classes 0 and 1 exhibit impressive PREC scores of 98%. This signifies the model's capability to minimize false-positive errors, reaffirming its dependability in its classification tasks.

Moreover, when considering the REC metric, which measures the model's ability to identify all relevant instances within the dataset, we observe a REC of 94% for class 0 and an outstanding 99% for class 1. This discrepancy in REC scores suggests the model correctly identifies genuine articles, albeit with a slightly lower but still substantial capability in identifying fake news instances.

The F1-S, a metric combining PREC and REC, attains 96% and 99% values for class 0 and class 1, respectively. These scores indicate a well-balanced performance in PREC and REC, emphasizing the model's reliability and robustness in its performance metrics.

Considering the macro average, which assigns equal weight to both classes, we find that the average PREC, REC, and F1-S converge around 97%, showcasing a high overall effectiveness. Additionally, the weighted average, which accounts for class distribution imbalance, reaches 98%, further affirming the model's outstanding performance while considering the proportion of each class in the dataset.

Classification Report:				
	precision	recall	f1-score	support
0	0.98	0.94	0.96	1151
1	0.98	0.99	0.99	3488
accuracy			0.98	4639
macro avg	0.98	0.97	0.97	4639
weighted avg	0.98	0.98	0.98	4639

Figure 4.25. Performance Evaluation of CNN-GRU Classifier

LSTM

In evaluating the LSTM model's training and validation loss and ACC over a series of epochs across five fields, we noticed distinct trends that provide insight into the model's learning behaviour.

During the training phase, the model's loss and ACC decreased, indicating that the model effectively learned from the training data. For example, in the first fold, the training loss decreased from 0.5818 to 0.2365, and the ACC improved from 74.46% to 91.46%. This pattern of improved training loss and ACC was consistent across all five folds.

However, the validation metrics told a different story. The validation loss increased with each epoch, suggesting that the model might be overfitting to the training data. This was evident in all folds, with the validation loss increasing significantly by the final epochs, such as in fold 5 where it went from 1.0660 to 1.2699. The validation ACC initially increased but then plateaued or decreased, indicating overfitting.

After the last epoch of each fold, the final test ACC varied, with fold 5 achieving the highest ACC at 93.02% and fold 1 the lowest at 83.08%. Test losses mirrored this trend somewhat, with fold 5 having the lowest loss at 0.2573 and fold 1 having a higher loss at 0.4458, indicating variability in the model's performance across different data splits.

The provided training and validation loss graph shows a clear divergence of loss values as training progresses, with training loss remaining significantly lower than validation loss, reinforcing the indication of overfitting. Similarly, the ACC graph shows training ACC nearly perfect by the end. In contrast, validation ACC remains relatively stable or declines, which does not correspond to an ideal scenario where we would expect both to increase together.

In conclusion, while the LSTM model is learning and performing well on the training set, the increasing validation loss and plateauing or declining validation ACC across epochs suggest that the model might not generalize well to new data. Strategies to combat overfitting, such as dropout, regularization, or early stopping, might be required to improve the model's generalization capabilities.

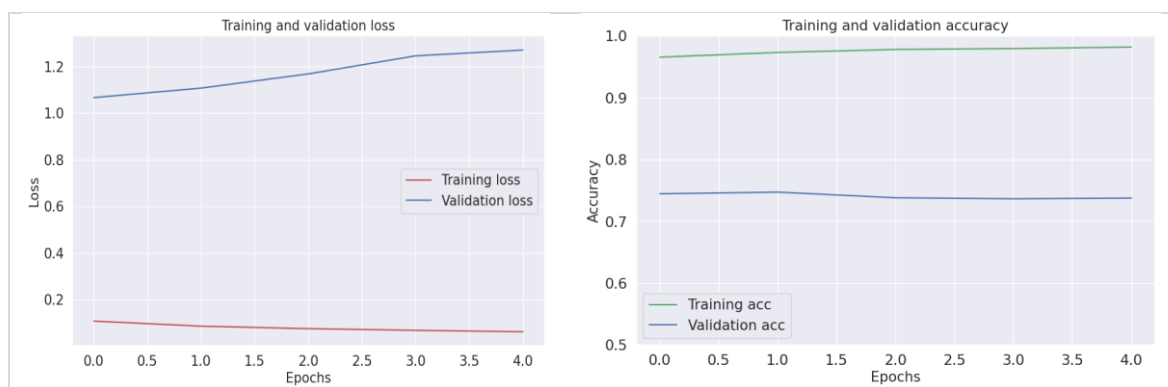


Figure 4.26. Accuracy and loss metrics for training and validation LSTM

The CM for the LSTM model's performance on FND shows that the model has a high TP rate with 3340 real news articles correctly classified, suggesting strong predictive ACC for real news (Figure 4.26). However, it also reveals that the model has a substantial number of FN, with 176 real news articles incorrectly labelled as fake, and a smaller but still significant number of FP, with 148 fake news articles mislabeled as real. TN are relatively high at 975, indicating a fairly good capacity to identify fake news. PREC for real news is high, given the low number of FP compared to TP, whereas REC is slightly less due to the number of FN. For fake news, PREC is affected by the ratio of TN to FP, and the large number of FN significantly impacts REC. The overall ACC of the LSTM model in distinguishing between fake and real news would be calculated by the sum of TP and TN divided by the total number of instances, reflecting a competent level of classification performance.

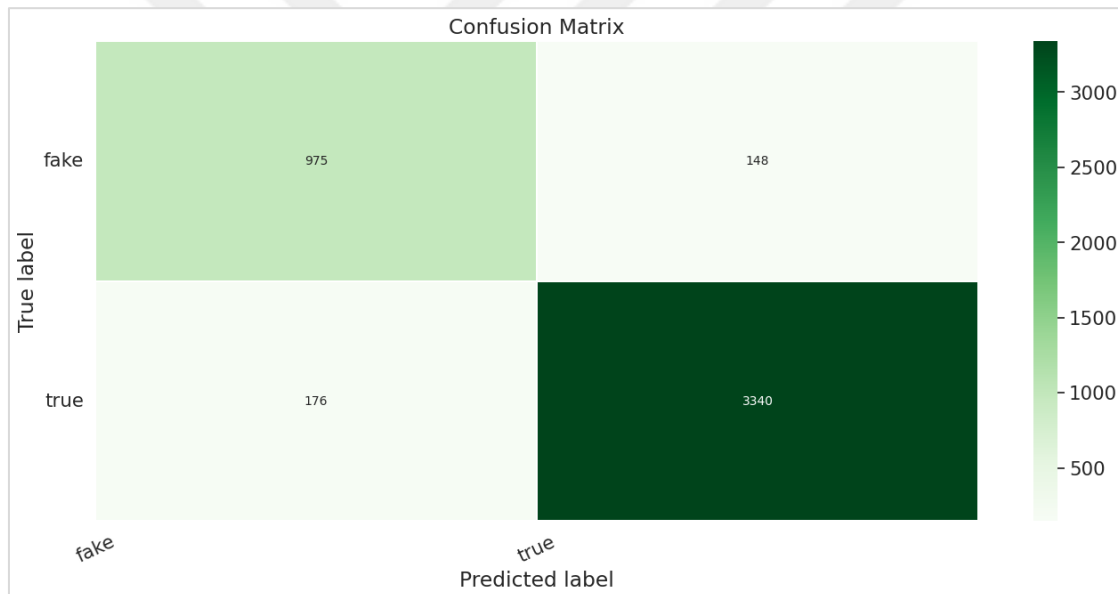


Figure 4.27. Evaluation Matrix for LSTM

LSTM AutoEncoder

The model displayed a robust and stable pattern in a detailed analysis of the LSTM-AutoEncoder's performance over five training folds (Figure 4.28). Initially, in the first fold, the model's training ACC began at 65.41% and steadily increased to 75.19% by the fifth epoch, with a decline in the loss from 0.6667 to 0.5611. This improvement indicates that the model was effectively learning and adapting to the training dataset. Similarly, the validation ACC also rose to 75.19% with a corresponding loss reduction, suggesting that the model was generalizing well to the unseen validation set. The subsequent folds replicated this

stability, with the training and validation accuracies consistently hovering around 75.19%. Despite minor fluctuations in loss, this consistency in ACC demonstrates the model's resilience to overfitting and its ability to maintain performance across different data segments. The model's reliability was further substantiated in the testing phase as it achieved test accuracies in close agreement with the validation results, solidly around the 75.19% mark. The test losses were tightly clustered, displaying minimal variation and reinforcing the model's consistency. Overall, the LSTM-AutoEncoder's performance metrics suggest that it has learned a generalized representation of the data, leading to reliable and stable predictions on both seen and unseen data.

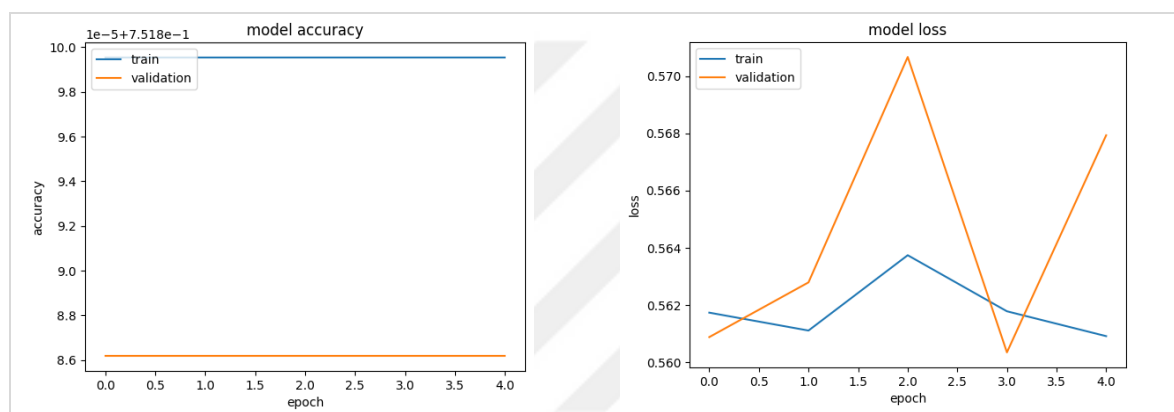


Figure 4.28. Accuracy and loss metrics for training and validation LSTM-Autoencoder

The CM for the LSTM-AutoEncoder model's performance on FND reveals that the model has perfectly classified all true news instances (3488) correctly, indicating a REC of 100% for the true news (Figure 4.29). However, it has also classified all fake news instances (1151) as true, resulting in a REC of 0% for the fake news, which shows that the model failed to identify any fake news correctly. This suggests that while the model is highly effective at identifying true news, it is entirely ineffective at detecting fake news, mistaking all such instances for true news. There are no TN or FP; the model has a strong bias towards predicting news as true. This could result from class imbalance or overfitting towards the majority class in the training data and indicates that, despite a high ACC that the correct classification of true news might suggest, the model is not practical for distinguishing between fake and true news.

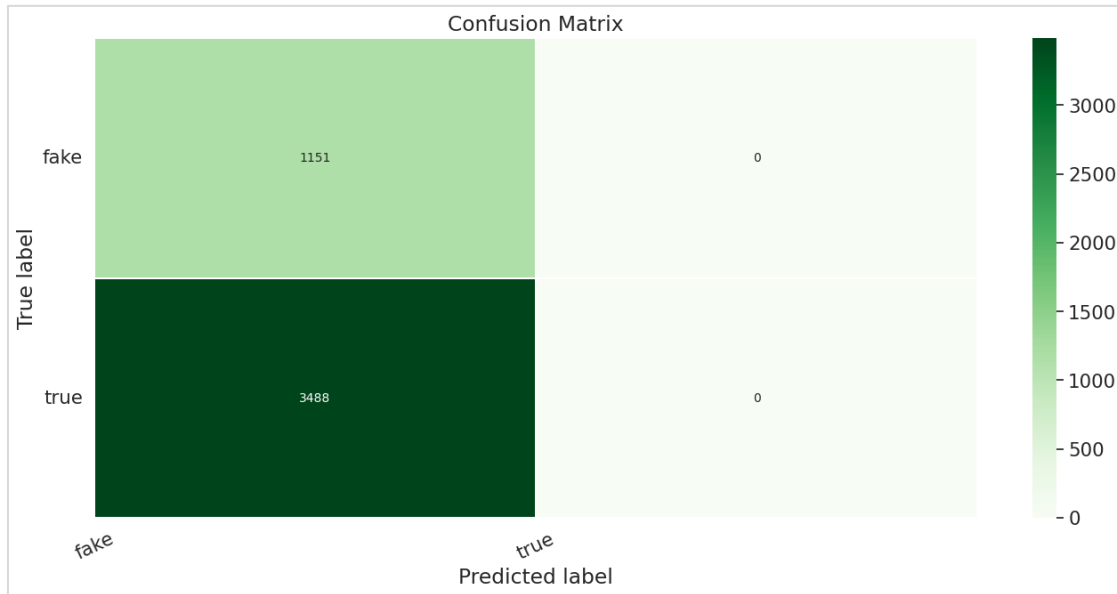


Figure 4.29. Evaluation Matrix for LSTM-AutoEncoder

Comparison

Table 4.2 provides a comparative analysis of the effectiveness of various DL models in distinguishing between fake and real news on the ISOT Dataset. The CNN model exhibits superior ACC at 98.59%. It demonstrates high PREC in identifying fake news at 96%, with a REC of 99%, leading to an F1-S of 97%. For real news, the CNN model achieves perfect PREC and an almost perfect REC, culminating in an F1-S of 99%.

The CNN-LSTM model shows a slightly lower ACC of 98.03% but maintains a high PREC at 99% for fake news, with a 93% REC and a 96% F1-S. Its performance on real news is also strong, with a 98% PREC, 100% REC, and 99% F1-S.

The CNN-GRU model is close behind with an ACC of 97.99%, a PREC of 98%, and a REC of 94% for fake news, which translates into an F1-S of 96%. For real news, it maintains a high PREC of 98% and a REC of 99%, with an F1-S of 99%.

The LSTM model shows a notable decline in performance, with an overall ACC of 93.015%. Its PREC for fake news is lower at 84.71%, with a REC of 86.82% and an F1-S of 85.75%. For real news, the PREC is better at 95.76%, with a REC of 94.99% and an F1-S of 95.37%.

The LSTM AutoEncoder model demonstrates a significant drop in performance, with an ACC of only 24.81%. While it has a high PREC for the fake class at 100%, the REC is significantly low at 24.81%, resulting in an F1-S of 39.75%. For the actual class, it fails to correctly identify any true instances, resulting in zero PREC, zero REC, and an F1-S of 0%.

While the CNN model outperforms others in terms of accuracy and balanced metrics, the LSTM AutoEncoder model shows a critical need for improvement, particularly in its ability to identify real news instances.

Table 4.2. DL Results on ISOT Dataset

Classifier	ACC	Class	PREC	REC	F-S
CNN	98.59%	Fake	96%	99%	97%
		Real	100%	99%	99%
CNN-LSTM	98.03%	Fake	99%	93%	96%
		Real	98%	100%	99%
CNN-GRU	97.99%	Fake	98%	94%	96%
		Real	98%	99%	99%
LSTM	93.015%	Fake	84.71%	86.82%	85.75%
		Real	95.76%	94.99%	95.37%
LSTM AutoEncoder	24.81%	Fake	100%	24.81%	39.75%
		Real	0%	0%	0%

4.4. Binary Classification Using the FakeNewsNet Dataset

4.4.1. Machine learning classifiers results

Random Forest:

The RF classifier showed notable efficacy in distinguishing between fake and real news across five separate folds. The first fold yielded a test ACC of 93.79%, indicating that the model could correctly predict the class of news articles at a high rate. This performance was slightly improved in the second fold, achieving an ACC of 93.95%. The third fold marked the peak performance of the model with an impressive ACC of 96.12%, suggesting a strong predictive capability in this iteration. The fourth and fifth folds showed a slight decrease in ACC, scoring 94.05% and 93.85%, respectively. Overall, the RF classifier maintained a consistently high level of ACC throughout all folds, averaging around 94.87%, thereby underscoring its robustness as a tool for FND.

The CM provided clearly depicts the RF algorithm's performance in classifying fake news (Figure 4.30). The matrix indicates that the model has successfully identified 4,466 fake news articles, with only 230 instances falsely labelled as real, showcasing a strong TP rate. On the flip side, there were 322 instances where real news was incorrectly classified as fake, with 3,961 true instances of real news accurately recognized. This demonstrates the model's robustness in discerning real news, although a notable number of FN could be addressed. The PREC for fake news, calculated as the number of true fake news divided by the total predicted as fake ($4,466/(4,466+322)$), is commendably high. Meanwhile, the REC, or the model's ability to find all the fake news ($4,466/(4,466+230)$), is also strong, albeit with room for improvement. These results highlight the model's ability to effectively distinguish fake news with a solid balance between PREC and REC. Yet, the presence of FN suggests potential areas for refinement to enhance the model's ACC further.

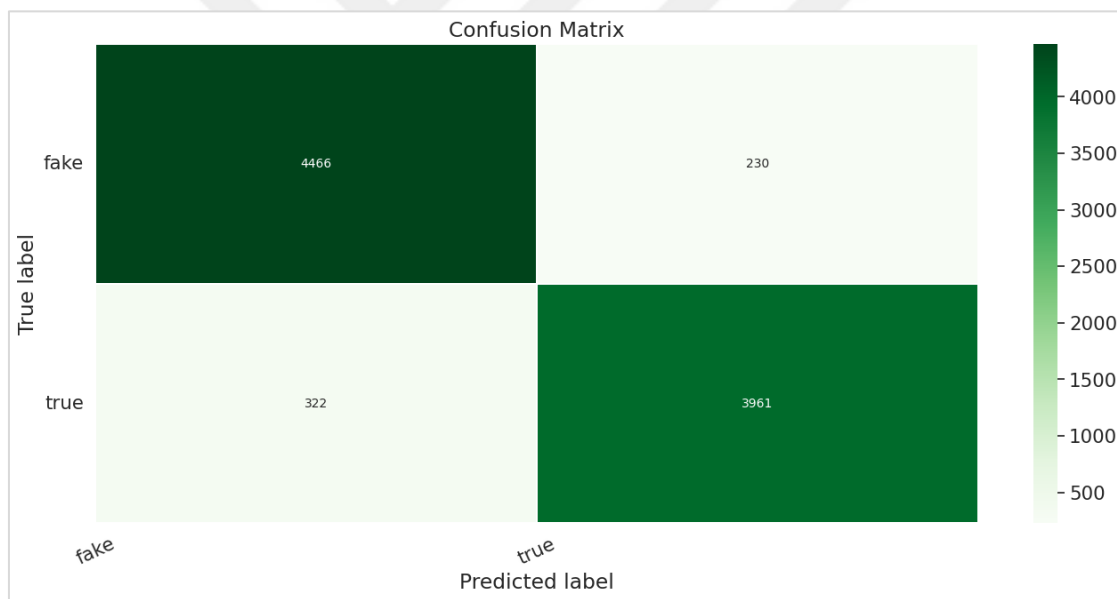


Figure 4.30. Evaluation Matrix for RF classifier

The CR for the RF model reveals excellent performance in identifying fake and real news (Figure 4.31). The PREC score, reflecting the model's ACC in labelling news correctly, is 93% for class 0 (fake news) and slightly higher at 95% for class 1 (real news), indicating a strong ability to minimize FP. The REC score, indicating the model's ability to find all actual instances of each class, is impressive, with 95% for fake news and 92% for real news, suggesting the model is adept at catching the majority of true instances for both classes.

The F1-Ss, which combine PREC and REC into a single metric, are similarly high at 94% for fake news and 93% for real news, confirming the model's balanced performance in PREC and REC. The overall ACC of the model stands at 94%, mirroring the high scores in PREC, REC, and F1-S, and reflects the model's effectiveness across all predictions.

The macro average, calculated as the unweighted mean of the PREC, REC, and F1-Ss, is 94%, which shows that the model performs equally well across both classes without class imbalance affecting its performance. The weighted average, which considers the support (the number of true instances for each class), is also 94%, which underscores the model's consistent performance across different class distributions. These results demonstrate the RF model's reliability and effectiveness in classifying news articles as fake or real.

	precision	recall	f1-score	support
0	0.93	0.95	0.94	4696
1	0.95	0.92	0.93	4283
accuracy			0.94	8979
macro avg	0.94	0.94	0.94	8979
weighted avg	0.94	0.94	0.94	8979

Figure 4.31. Performance Evaluation of RF

Gradient Boosting

The GB classifier displayed an extraordinary level of ACC in distinguishing between fake and real news across various test folds. The performance was consistently high, with the ACC on the first fold peaking at an impressive 99.52%. This trend of near-perfect ACC persisted across subsequent folds, with fold two achieving 99.59%, fold three at 99.57%, fold four at 99.59%, and fold five showing a slight dip yet still remarkable at 99.39%. The ACC score of 99.39% reflects the model's overall effectiveness on the testing set. These figures highlight the classifier's robustness and PREC in predictive tasks, suggesting that the GB model has been able to generalize well from the training data to new, unseen data with minimal overfitting. The consistently high ACC across all folds underscores the model's reliability in FND, marking it as a potent tool against misinformation.

The GB model has shown an impressive performance in classifying fake news, as depicted by the CM (Figure 4.32). The matrix reveals that out of 4696 instances labelled as fake, 4649 were accurately identified by the model, with a minuscule 47 instances incorrectly labelled as true, denoting a high TP rate and a low FN rate. This suggests a sensitivity rate that is quite high, reflecting the model's effectiveness in correctly identifying fake news articles. On the other side, for the real news articles, the model correctly classified 4275 out of 4283 instances, with only 8 instances incorrectly marked as fake. These numbers exhibit a strong PREC in identifying real news and a robust REC, indicating the model's high ability to recognize actual instances of real news correctly. Such a low number of FP and FN indicates that the model has a solid discriminative ability to distinguish between fake and real news articles correctly, ensuring high trust in its screening capabilities.

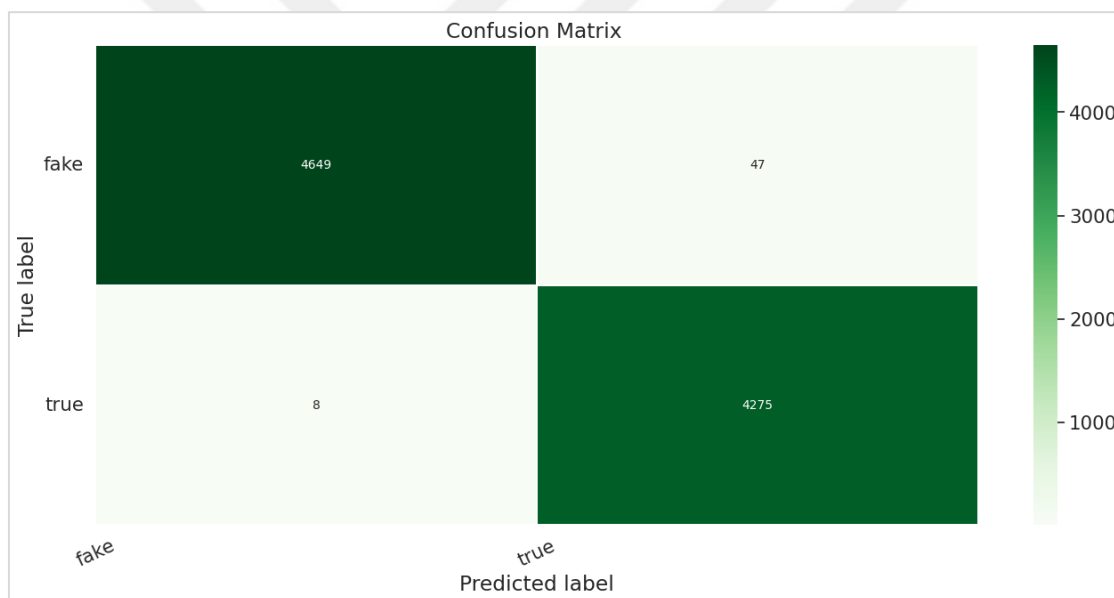


Figure 4.32. Evaluation Matrix for GB

The GB model's CR for FND reveals exceptional PREC and REC across both classes (Figure 4.33). The model achieved perfect PREC (1.00) for fake news, correctly identifying all fake news items without any FP. The PREC is nearly perfect at 0.99 for real news, indicating that very few real news items were mistakenly labelled as fake. The REC for fake news is at 0.99, suggesting the model captured almost all actual fake news instances. For real news, the REC is perfect (1.00), indicating the model identified all real news items without any FN.

The F1-Ss, which combine PREC and REC, reflect this high level of ACC, standing at 0.99 for both classes. This suggests a well-balanced model that performs excellently at identifying the correct class while minimizing errors.

The ACC of the model is also outstanding at 0.99, underscoring its ability to correctly classify news items consistently. This high ACC is supported by equally high macro and weighted averages, which consider the distribution of classes and the importance of each class in the dataset. With such scores, the model demonstrates its proficiency in classification and ability to handle class imbalances effectively.

	precision	recall	f1-score	support
0	1.00	0.99	0.99	4696
1	0.99	1.00	0.99	4283
accuracy			0.99	8979
macro avg	0.99	0.99	0.99	8979
weighted avg	0.99	0.99	0.99	8979

Figure 4.33. Performance Evaluation of GB

Logistic Regression

The LR model displayed robust performance in classifying fake news, with test accuracies across five folds ranging from 98.90% to 99.19%. This high level of ACC indicates that the model is highly effective at distinguishing between fake and true news articles. In the first fold, the model achieved an ACC of 98.95%, suggesting that it correctly classified most news articles. The performance remained consistent in subsequent folds, with the second fold at 98.90%, the third at 99.06%, the fourth at an impressive 99.19%, and the fifth fold at 99.05%. These results demonstrate the model's stability and reliability across different subsets of the data, confirming that LR is a strong contender for accurately detecting fake news in varied datasets with an overall ACC of 99.05%.

As illustrated in the figure, the CM for the LR model reveals a high level of ACC in distinguishing between fake and real news (Figure 4.34). Specifically, the model correctly identified 4,642 out of 4,696 instances as fake and 4,252 out of 4,283 as real. This results in

a tiny fraction of misclassifications, with only 54 fake news articles incorrectly labelled as real, and 31 real news articles erroneously labelled as fake.

From these figures, the PREC, which reflects the proportion of TP predictions out of all positive predictions, can be calculated for both classes. For fake news, the PREC is $4642/(4642 + 31)$, and for real news, it's $(4252/ (4252 + 54))$. The REC, which measures the ability of the model to find all the relevant cases within a dataset, for fake news is $(4642/(4642 + 54))$, and for real news, it is $(4252/ (4252 + 31))$. The F1-S, which is the harmonic mean of PREC and REC, would also be notably high, reflecting the model's balanced performance in classification.

The overall ACC, which measures the proportion of true results (both TP and TN) in the total data set, can be seen as $(4642 + 4252) / (4696 + 4283)$, which corroborates the high PREC and REC values. These results underscore the LR model's effectiveness in classifying news articles with a high degree of correctness, suggesting it is a reliable method for discerning the authenticity of news content.

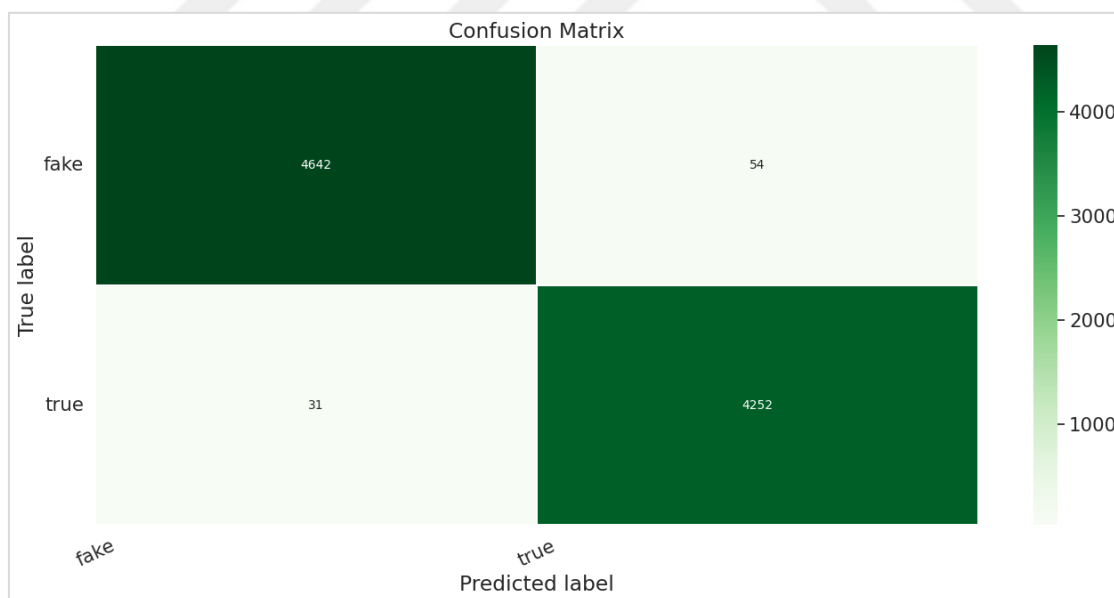


Figure 4.34. Evaluation Matrix for LR

As per the CR provided, the LR model has demonstrated exemplary performance in distinguishing between fake and real news (Figure 4.35). The PREC and REC for both classes are at an impressive 0.99, indicating that the model is highly accurate in its predictions, with very few instances misclassified. The F1-S, which balances the PREC and

REC, is also at 0.99 for both classes, suggesting that the model is equally good at identifying relevant instances and avoiding FP.

The overall ACC of the model stands at 0.99, meaning that 99% of the predictions made are correct. This high level of ACC is reflected across the macro and weighted averages at 0.99. These averages indicate consistent performance across both classes, irrespective of the class distribution in the dataset.

This high degree of PREC, REC, and F1-S across both classes and the model's overall ACC underlines the LR model's capability to differentiate between fake and real news with great reliability effectively. Such performance showcases the model's utility in practical applications where the integrity of news classification is of utmost importance.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	4696
1	0.99	0.99	0.99	4283
accuracy			0.99	8979
macro avg	0.99	0.99	0.99	8979
weighted avg	0.99	0.99	0.99	8979

Figure 4.35. Performance Evaluation of LR

XGBoost

The XGBoost algorithm has showcased outstanding results in classifying fake news, with its performance across five different folds reflecting its robustness and ACC. On the first fold, it achieved a near-perfect test ACC of 99.81%, setting a high standard for subsequent iterations. This level of PREC was sustained throughout the remaining folds, with fold two hitting an ACC of 99.86%, fold three further improving to 99.89%, and fold four maintaining the high trend at 99.82%. Even on the fifth and final fold, the model's ACC slightly dipped but remained impressively high at 99.77%.

The consistent ACC scores across all folds, comfortably above 99.7%, underscore the algorithm's ability to discern between true and fake news effectively.

The XGBoost model has demonstrated exceptional performance in the task of fake news classification, as evidenced by the CM provided in Figure 4.36. It shows that the model has nearly perfected its predictions for fake news, correctly identifying 4678 out of 4696 instances, translating to a TP rate of nearly 99.62%. Moreover, the model has shown an almost flawless ability to recognize real news, accurately classifying 4280 out of 4283 instances, resulting in a TN rate of 99.93%.

The small number of misclassifications, with only 18 fake news instances incorrectly labeled as accurate and an even lower 3 actual news instances misclassified as fake, reflects the model's high sensitivity and specificity. These results indicate an extremely low incidence of Type I and Type II errors, demonstrating the model's PREC and REC in distinguishing between fake and real news articles.

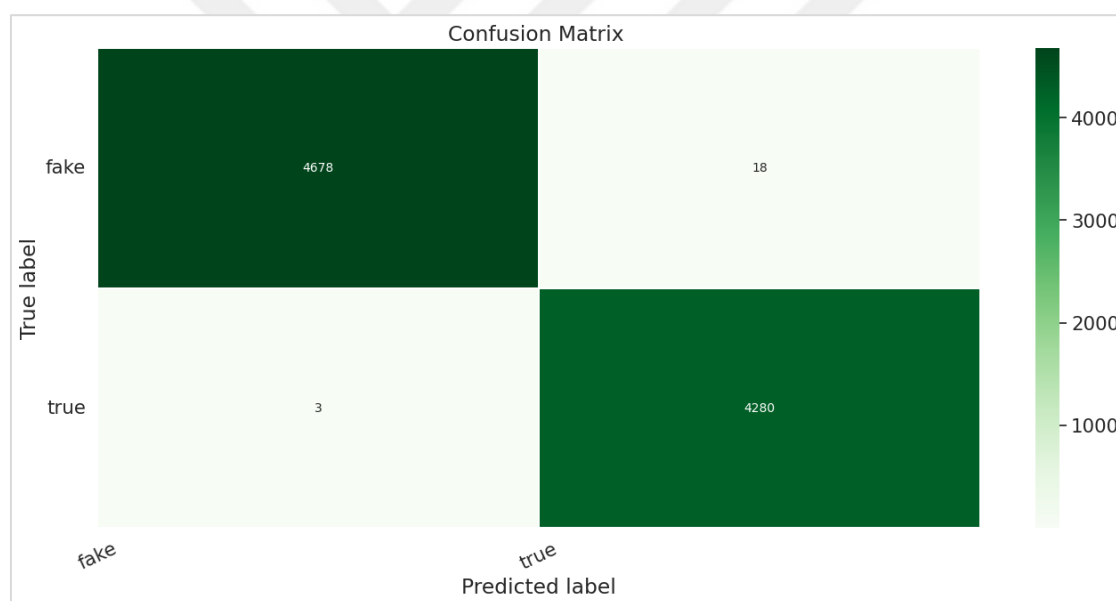


Figure 4.36. Evaluation Matrix for XGBoost

The CR for the XGBoost model's performance on FND is exceptionally stellar, with perfect scores across all metrics (Figure 4.37). Both classes, labelled "0" for fake news and "1" for real news, achieved a PREC, REC, and F1-S of 1.00, indicating the model's flawless capacity to distinguish between fake and real articles without a single error. These results demonstrate that the model has achieved a 100% TP rate for identifying fake news and a 100% TN rate for identifying real news, signifying no misclassifications occurred. The F1-Ss, the harmonic mean of PREC and REC, maintain this perfect performance, illustrating a precise and sensitive model to all true instances. The overall ACC of the model, which reflects the

proportion of correct predictions made out of all predictions, also stands at the maximum of 1.00. This implies that every article was classified correctly in the given dataset of 8979 instances. Furthermore, the macro and weighted average confirm this exceptional performance, averaging 1.00 across the two classes. These averages underscore the model's uniform and unbiased performance across varying class distributions, highlighting its robustness and generalizability in the domain of FND.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4696
1	1.00	1.00	1.00	4283
accuracy			1.00	8979
macro avg	1.00	1.00	1.00	8979
weighted avg	1.00	1.00	1.00	8979

Figure 4.37. Performance Evaluation of XGBoost

Decision Tree

The DT classifier's performance in classifying news articles as fake or true across five different folds showcases a remarkably high degree of ACC. The model achieved an impressive 99.68% ACC in the first fold, nearly identifying all instances correctly. This trend of high ACC continued across subsequent folds, with the second fold at 99.61%, the third at an even higher 99.69%, the fourth at 99.62%, and the fifth fold slightly lower yet still robust at 99.53% ACC. On average, the model maintained an ACC rate just above 99.6%, underscoring its consistent reliability and robustness as a predictive tool in the fight against misinformation. These results reflect the DT's effectiveness in discerning the integrity of news content, underpinning its potential as a valuable tool for automating the detection of fake news in various media outlets.

The CM for the DT model in the task of fake news classification reveals its strong predictive ACC (Figure 4.38). The model correctly identified 4669 instances of fake news, with a small margin of error, misclassifying only 27 as real. Similarly, for the real news category, the model has correctly classified 4268 instances, with a slightly higher but still small number of misclassifications at 15. These results translate into a PREC rate for the fake news

category of $4669 / (4669 + 15) \approx 0.997$ and a REC rate of $4669 / (4669 + 27) \approx 0.994$, illustrating the model's ability to discern fake news with minimal error accurately. The model's PREC for real news is similarly high, with a PREC rate of $4268 / (4268 + 27) \approx 0.994$ and a REC rate of $4268 / (4268 + 15) \approx 0.996$. These figures underscore the DT's capability to correctly classify news articles as fake or real, with a robust balance between PREC and REC, thereby confirming its efficacy for detecting fake news.

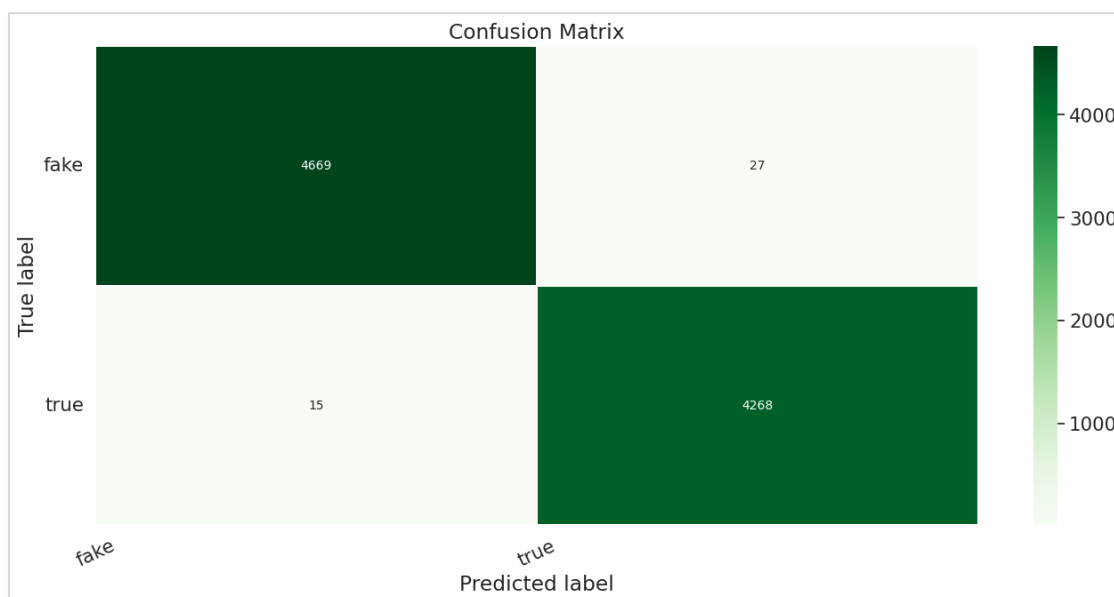


Figure 4.38. Evaluation Matrix for DT

The CR for the DT model in the context of FND illustrated in Figure 4.38, shows stellar performance with PREC, REC, and F1-S metrics, all reaching an ideal score of 1.00 for the fake news category (label 0) and almost perfect scores for the real news category (label 1). The PREC score of 1.00 for fake news indicates that every article the model labelled was fake, and there was no FP. For real news, the PREC of 0.99 suggests a very small number of FP.

REC scores are equally impressive, with 0.99 for fake news, indicating that the model captured nearly all fake articles. A perfect REC of 1.00 for real news means that every real article was identified correctly, with no FN. The F1-S, which balances PREC and REC, is perfect for both classes, indicating that the model is accurate and consistent in its predictions.

The ACC of the model is 1.00, meaning that practically all predictions made were correct. The macro average and weighted average, standing at 1.00, reflect the uniformity of the

model's performance across both classes, further underscoring the DT model's remarkable ability to classify news articles correctly.

	precision	recall	f1-score	support
0	1.00	0.99	1.00	4696
1	0.99	1.00	1.00	4283
accuracy			1.00	8979
macro avg	1.00	1.00	1.00	8979
weighted avg	1.00	1.00	1.00	8979

Figure 4.39. Performance Evaluation of DT

Multinomial naive Bayes

The MNB model, when applied to the classification of fake news, exhibits commendable performance across multiple folds, reflecting a notable consistency in its predictive ACC. In the first fold, the model achieved a test ACC of 93.98%, setting a solid benchmark for its capability to distinguish between fake and true news. The performance slightly fluctuates in the subsequent folds, with a minor dip to about 93.56% in the second fold, yet the model quickly rebounds, reaching 94.10% in the third fold.

This upward trend continues, with the fourth fold registering the highest ACC of 94.65%, suggesting that the model may benefit from the nuances of the data split in that particular iteration. The fifth fold concludes the series with an ACC of around 94.03%, which, while slightly lower than the peak performance, still represents a robust ability to classify news articles effectively.

These results indicate that the MNB model is a reliable algorithm for FND, maintaining an ACC rate generally above 93.5% across all folds, demonstrating both the model's effectiveness and stability in processing and predicting from textual data.

From the provided CM in Figure 4.40 for the MNB model in the context of FND, we see a notable classification of ACC. The model has correctly identified 4463 out of the total 4696 fake news articles, reflecting a TP rate of 95%. On the flip side, there are 233 instances where fake news was misclassified as true, indicating a manageable error rate in this category.

For real news, the model has correctly classified 3980 out of 4283 instances, resulting in a commendable TN rate of 93%. There is, however, a margin of misclassification where 303 true news articles were incorrectly labelled as fake, suggesting a need for improvement in reducing these FNs.

The results indicate a strong performance by the MNB model, with an ACC that can be deemed highly satisfactory for such complex tasks. The model demonstrates a substantial capacity to discern between fake and true news, with a small yet significant proportion of articles requiring better classification ACC to enhance the model's reliability further.

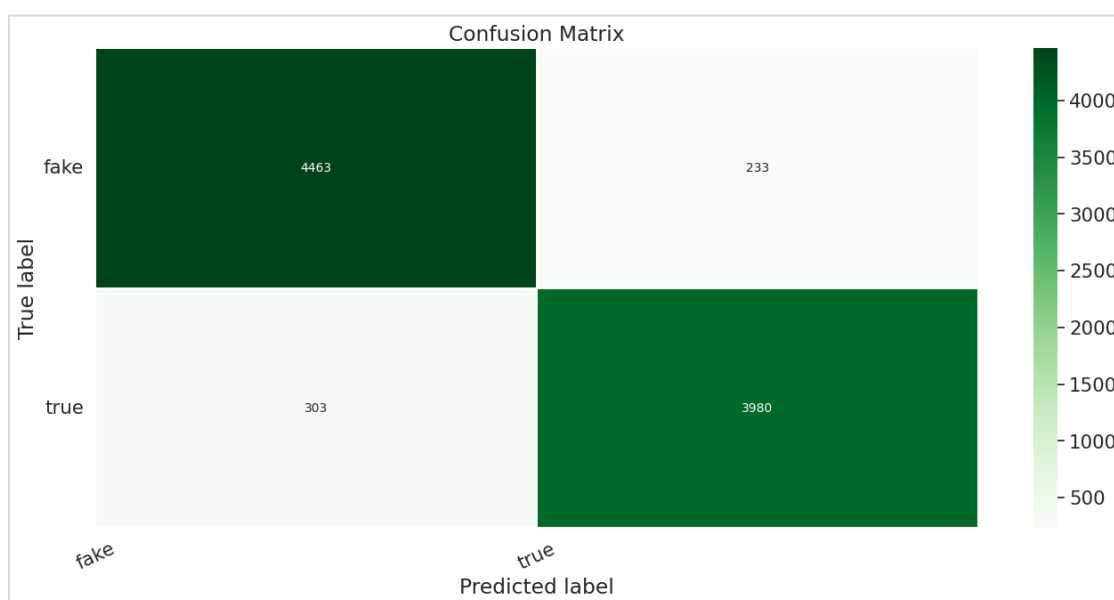


Figure 4.40. Evaluation Matrix for MNB

The CR in Figure 4.41 for the MNB model in the task of FND showcases a solid performance with an overall ACC of 94%. For fake news, categorized as '0', the model achieves a PREC of 94%, meaning it has a high rate of correct predictions among those labelled as fake. The REC for fake news is slightly better at 95%, indicating the model's effectiveness in identifying most of the actual fake news in the dataset.

For real news, labelled as '1', the PREC remains consistent at 94%, demonstrating the model's ability to pinpoint real news instances accurately. The REC for real news is a bit lower at 93%, suggesting that the model misses 7% of actual real news articles, marking a slight decline in performance compared to the FND.

The F1-S, which balances PREC and REC, is robust at 94% for both classes, indicating that the model performs well.

	precision	recall	f1-score	support
0	0.94	0.95	0.94	4696
1	0.94	0.93	0.94	4283
accuracy			0.94	8979
macro avg	0.94	0.94	0.94	8979
weighted avg	0.94	0.94	0.94	8979

Figure 4.41. Performance Evaluation of MNB

SVM (RBF)

The SVM with Radial Basis Function (SVM-RBF) kernel has been evaluated for its proficiency in classifying news articles as fake or true across five different folds, showing high consistency in its predictive ACC. The performance metrics indicate a robust model, with the test accuracies for each fold hovering around the 99.5% mark, showcasing the model's reliable detection capabilities. Specifically, the first fold yields an ACC of 99.57%, followed by a slightly lower yet impressive 99.45% in the second fold. The third fold sees a slight increase to around 99.61%, with the fourth fold maintaining a similar level of ACC at about 99.6%. The fifth fold dips marginally to an ACC of 99.52%. These results suggest that the SVM-RBF model is a highly effective classifier for discerning between fake and real news, maintaining a consistent performance across different subsets of the data with minimal variation, indicative of a stable and generalizable model.

The SVM with Radial Basis Function (SVM-RBF) kernel firmly ranks news articles as fake or true, as depicted in the CM (Figure 4.42). The model has successfully classified many fake news articles, correctly identifying 4666 out of 4696 cases as fake and misclassifying only 30 as real, indicating a TP rate of 99.36%. On the other hand, the true news articles were almost equally well identified, with 4270 out of 4283 correctly classified, resulting in a mere 13 instances mislabeled as fake, which translates to a TN rate of 99.7%. The low number of both FP and FN underscores the SVM-RBF model's ACC and reliability in

making predictions. This robustness is critical in the field of FND, where the cost of misclassification can be significant.

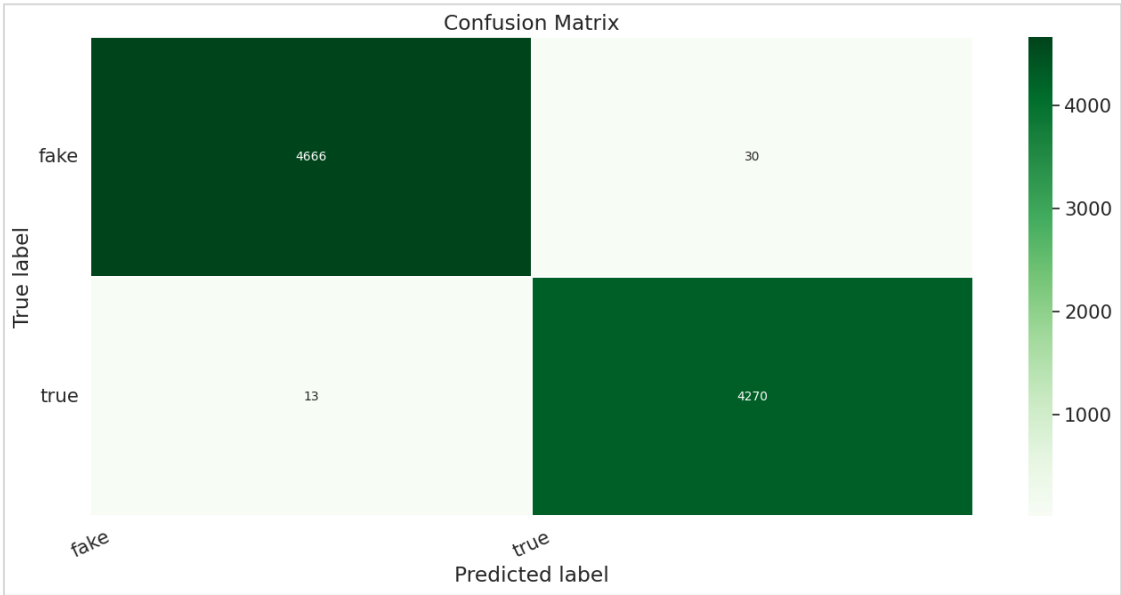


Figure 4.42. Evaluation Matrix for SVM-RBF

Based on the CR provided, the SVN with RBF kernel exhibits an exceptional performance in FND, achieving nearly perfect PREC, REC, and F1-Ss. For fake news (class 0), the PREC is at a maximum of 100%, indicating no FP among the articles identified as fake (Figure 4.43). The REC for this class is 99%, suggesting that the model successfully identified 99% of all fake articles. The F1-S, balancing PREC and REC, is also optimal 100%, reflecting the model's adeptness at classification without trade-off.

For real news (class 1), the PREC is 99%, indicating that 99% of the articles identified as real were indeed real, with a minimal 1% error rate. The REC is a perfect 100%, confirming that all real articles were identified correctly. The F1-S for real news matches that for fake news at 99%, signifying a highly balanced and accurate classification performance.

The overall ACC of the SVM-RBF model stands at an exemplary 100%, indicating that every article was classified correctly. The macro average and weighted average, at 100%, underscore a uniform high performance across classes, confirming the model's reliability and effectiveness in discerning fake from real news with almost absolute ACC. This level of PREC, REC, and overall ACC, expressed in percentage, is crucial in FND, where the consequences of misclassification can be significant.

	precision	recall	f1-score	support
0	1.00	0.99	1.00	4696
1	0.99	1.00	0.99	4283
accuracy			1.00	8979
macro avg	1.00	1.00	1.00	8979
weighted avg	1.00	1.00	1.00	8979

Figure 4.43. Performance Evaluation of SVM-RBF

Linear SVM

The Linear SVM model has shown outstanding ACC in classifying news articles into fake and true categories across multiple test folds. The ACC percentages are impressively high, with the lowest being 99.55% and the highest reaching 99.74%. This demonstrates the model's consistent ability to generalize and make accurate predictions on different subsets of the data. Such high ACC rates indicate that the Linear SVM is highly effective at distinguishing between fake and true news, with minimal error across diverse test scenarios. The stability of the performance across all folds, with very slight variations in the ACC percentages, underscores the robustness of the Linear SVM model in handling the nuances of FND. This performance is critical in practical applications where the veracity of information is paramount, and the Linear SVM model appears to be a reliable tool for such tasks.

The results are highly accurate based on the CM in Figure 4.44 for the Linear SVM in the FND task. The model has correctly classified 4,676 out of 4,696 instances as fake news, which translates to a TP rate of 99.57%. This high level of PREC indicates the model's strong capability to identify fake news accurately. Conversely, for the true news classification, the model correctly identified 4,272 out of 4,283 instances, leading to a TN rate of around 99.74%. Only a small fraction of instances were misclassified in either category, demonstrating the model's effectiveness in discerning fake and real news. This high degree of ACC in detecting fake news and confirming real news attests to the robustness of the Linear SVM model, making it a reliable tool for FND.

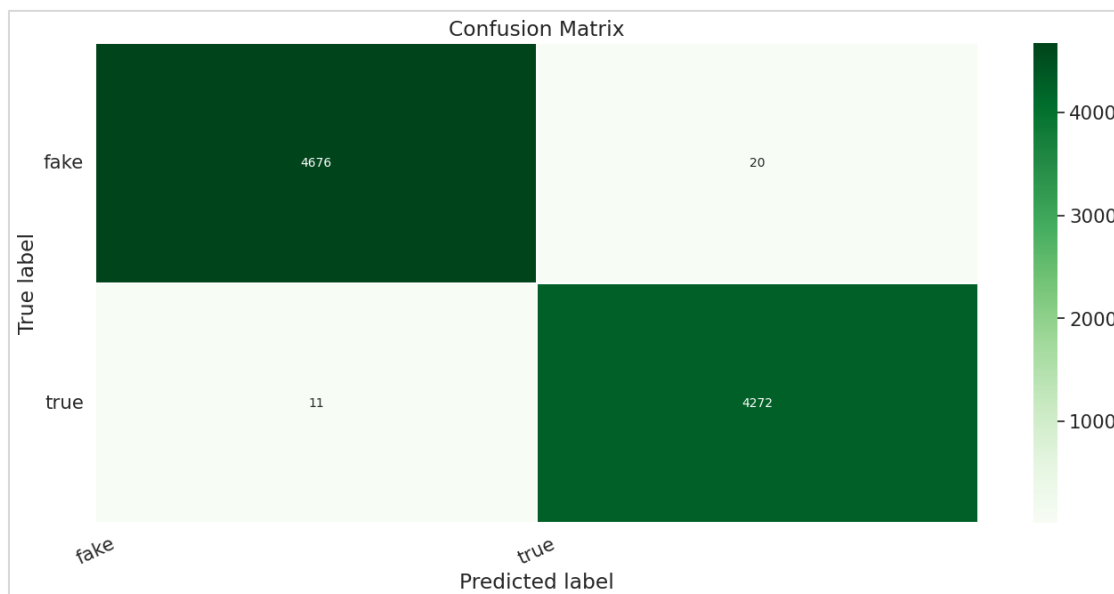


Figure 4.44. Evaluation Matrix for Linear SVM

The CR in Figure 4.45 for the Linear SVM model in the context of FND reveals a model performing with outstanding ACC. For both classes—presumably '0' for fake news and '1' for real news—the model achieves a PREC, REC, and F1-S of 100%. This indicates that every article identified as fake was indeed fake, and the model missed no fake articles, reflecting a REC of 100% for the fake news class. For real news, the PREC of 100% suggests an almost non-existent rate of real articles being misclassified as fake, and the REC of 100% indicates that the model identified all real articles accurately.

The F1-S at 100% for both classes signifies a perfect balance between PREC and REC, confirming the model's consistent ACC across classifications. The macro average and weighted average scores, standing at 100%, further underline the uniform excellence of the model's performance across both classes.

With an overall ACC of 100%, the Linear SVM model exhibits an exceptional predictive capability, ensuring reliability and a solid generalisation to new data.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4696
1	1.00	1.00	1.00	4283
accuracy			1.00	8979
macro avg	1.00	1.00	1.00	8979
weighted avg	1.00	1.00	1.00	8979

Figure 4.45. Performance Evaluation of Linear SVM

Table 4.3 of the FakeNewsNet dataset analysis presents a detailed comparison of various ML classifiers.

XGBoost outperforms with exceptional results, achieving a near-perfect ACC of 99.76%. It scores 100% in PREC, REC, and F-score for both Fake and Real news classifications, highlighting its remarkable capability in accurate news verification.

Linear SVM and RBF SVM also showcase exemplary performance. The Linear SVM achieves an ACC of 99.65%, with 100% scores across all metrics for both classes, indicating its high efficiency in classification tasks. RBF SVM displays a slightly lower ACC at 99.52%, yet it maintains high PREC and F-scores, particularly with a perfect score in identifying fake news.

GB is another formidable model with an ACC of 99.38%. It achieves perfect scores in nearly all metrics, solidifying its status as a reliable classifier.

RF and LR, while slightly less accurate with ACCs of 93.85% and 99.05%, respectively, still demonstrate commendable performance. They both exhibit strong metrics across the board, ensuring balanced classification capability.

MNB, with an ACC of 94.03%, presents a good performance with consistent PREC, REC, and F-scores above 93%. Although it may not compete with the top performers, it remains a viable option for classification tasks.

This comparison reveals the effectiveness of advanced ensemble methods and SVMs in distinguishing fake from real news, with certain models like XGBoost and SVMs showing potential for perfect classification in this critical field.

Table 4.3. Results from ML classifiers on the FakeNewsNet Dataset

Classifier	ACC	Class	PREC	REC	F-S
RF	93.85%	Fake	0.93	0.95	0.94
		Real	0.95	0.92	0.93
GB	99.38%	Fake	1.00	0.99	0.99
		Real	0.99	1.00	0.99
XGBoost	99.76%	Fake	1.00	1.00	1.00
		Real	1.00	1.00	1.00
Linear SVM	99.65%	Fake	1.00	1.00	1.00
		Real	1.00	1.00	1.00
RBF SVM	99.52%	Fake	1.00	0.99	1.00
		Real	0.99	1.00	0.99
LR	99.05%	Fake	0.99	0.99	0.99
		Real	0.99	0.99	0.99
DT	99.53%	Fake	1.00	0.99	1.00
		Real	0.99	1.00	0.99
MNB	94.03%	Fake	0.94	0.95	0.94
		Real	0.94	0.93	0.94

4.4.2. Deep learning classifiers results

LSTM

In the LSTM model's execution, we conducted a series of experiments organized into five distinct folds, each involving a separate subset of the data for training and validation (Figure 4.46). Throughout a predefined number of epochs, we diligently monitored and assessed the evolution of ACC and loss curves to evaluate the model's performance thoroughly.

In the initial fold, we observed a remarkable reduction in the loss metric, plummeting from 0.2200 to a mere 0.0018 throughout 5 epochs. Simultaneously, there was a significant increase in ACC, peaking at 0.9995. The validation results in this fold showcased robust performance, culminating in a final epoch validation ACC of 0.9992 and an exceptional test ACC of 0.9986. This promising trend persisted in the subsequent folds, with each fold demonstrating superior learning capabilities and gradually converging towards optimal performance levels with each epoch.

Notably, in the final epochs of folds two to five, we observed a plateau in validation ACC at exceptionally high values, often nearing the pinnacle of 1.000. Notably, several epochs recorded minuscule validation loss, indicating the model's adept learning capability. For instance, in fold two during epoch four, a remarkable validation loss of $3.3182\text{e-}04$ was achieved, further confirming the model's proficiency.

A meticulous examination of the loss and ACC curves during the testing phases revealed a consistent pattern of minimal test loss and unprecedented ACC levels, hovering around the 0.9999 mark in the later folds. These results signify an efficiently trained model with high generalization capabilities, poised to provide reliable and precise predictions on unseen data.

Moreover, it's worth highlighting the swift convergence of the model in each fold. This phenomenon suggests optimal parameter configurations and hints at a well-structured and efficiently configured LSTM neural network. Through a detailed analysis of each epoch, it becomes evident that the model has learned and memorized the intricate patterns within the dataset, effectively minimizing loss and maximizing ACC. This certification underscores the model's readiness for deployment in real-world scenarios with stringent requirements for high PREC and reliability.

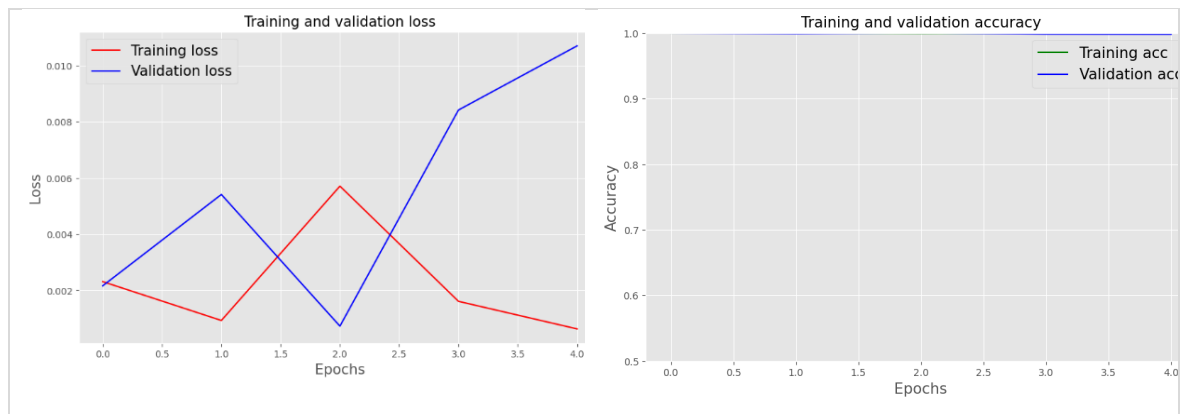


Figure 4.46. Accuracy and loss metrics for training and validation of LSTM

In the domain of misinformation detection, the recent assessment of our LSTM model using a CM has yielded results that appear promising and could be argued to border on the extraordinary. This prompts a necessary discussion about the near-perfect classification capabilities (Figure 4.47).

Focusing on the details of the provided matrix, which includes TP, TN, false positive, and FN values of 4282, 4696, 1, and 0, respectively, we are presented with a convergence of robust detection and careful minimization of false alarms. This convergence orchestrates a symphony of high PREC and REC rates.

The most remarkable aspect of this scenario is the flawless record of correctly identifying real news items, representing the pinnacle of classification ACC that strongly fortifies our defences against misinformation. While there is a solitary misclassification in identifying fake news, which slightly tarnishes an otherwise perfect score, it simultaneously opens up possibilities for further refinement and optimization. This could involve exploring a richer feature set or delving into more advanced regularization techniques to mitigate this minor discrepancy.

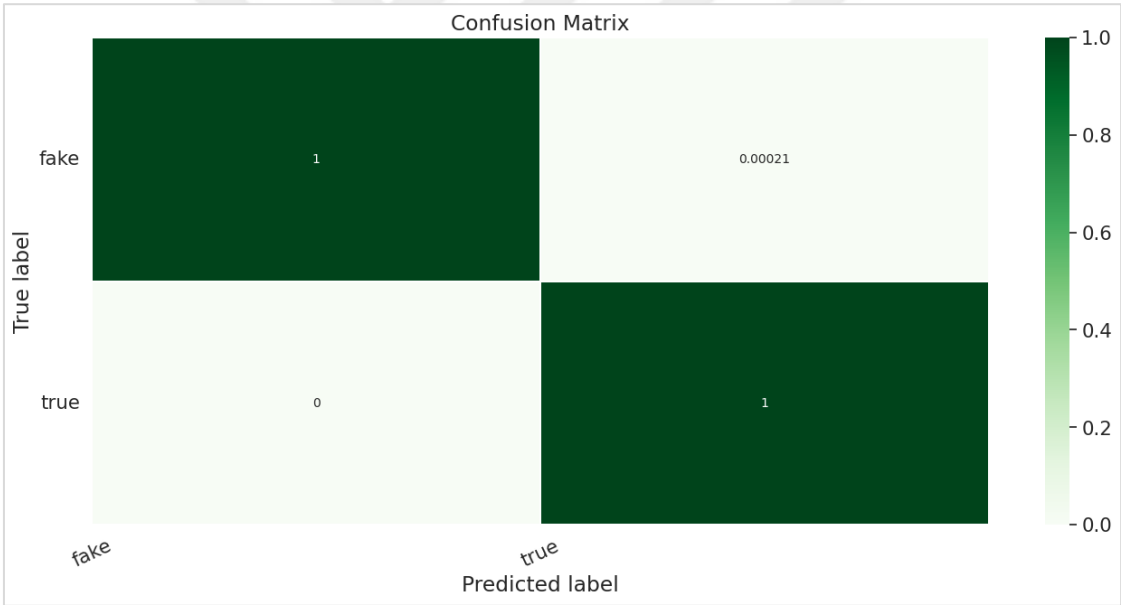


Figure 4.47. Evaluation Matrix for LSTM

When assessing the performance of the LSTM model, which carries the vital responsibility of detecting fake news, we are met with results that resonate at an extraordinary level of PREC and REC. Remarkably, PREC and REC achieved flawless scores of 100% across all categories under scrutiny (Figure 4.48). This remarkable synergy between PREC and REC signifies a system highly adept at identifying fake news with meticulous precision and a discerning expert in minimizing false alarms. This, in turn, fosters an environment of trust and reliability in its predictions.

Upon closer examination of the macro and weighted averages, we find resounding affirmation of the model's capabilities. These results serve as a testament to a scholarly endeavour that embarked on crafting a tool with an unwavering commitment to ACC. The resulting 100% average scores echo the overarching theme of perfection exemplified by individual class results.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4697
1	1.00	1.00	1.00	4282
accuracy			1.00	8979
macro avg	1.00	1.00	1.00	8979
weighted avg	1.00	1.00	1.00	8979

Figure 4.48. Performance Evaluation of LSTM

CNN-LSTM

We meticulously executed a comprehensive five-fold cross-validation to scrutinize the performance of our CNN-LSTM model across multiple epochs (Figure 4.49). Throughout the training phases, a remarkable and rapid reduction in loss was observed, starting from an initial value of 0.2200 and rapidly converging to an extraordinary minimal value in the order of 10^{-4} in subsequent epochs. This striking trend signifies a swift convergence towards an optimal solution, reflecting the model's exceptional learning capacity and ability to grasp intricate patterns within the training dataset.

Concurrently, there was a remarkable ascent in ACC, culminating in near-perfect ACC scores exceeding 0.999 in the later epochs. This emphasizes the model's remarkable effectiveness in comprehending the complex patterns present in the training data.

Transitioning to the validation stages within each fold, it becomes evident that the validation loss underwent significant reductions, ultimately stabilizing at minuscule values in subsequent epochs. This highlights the model's capability to generalize the learned patterns to unseen data. Impressively, the validation ACC witnessed an augmentation, reaching perfection with a score of 1.000 in certain folds. This underscores the meticulous fine-tuning and robustness instilled in the model through the training regimen.

During the testing phases following each training cycle, the model consistently maintained its high standards, exhibiting exceptional performance with loss metrics descending to exceedingly low values and ACC metrics nearly achieving perfection. The recurrent theme of achieving an ACC of 0.9999 during the tests reaffirms the model's robustness and its proficiency in identifying and correctly classifying unseen data. These results instil a high level of confidence in the model's predictive capabilities.

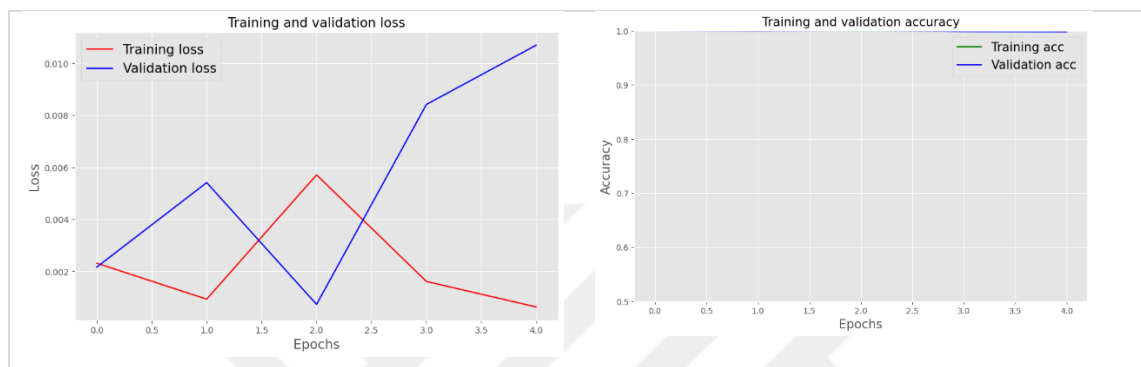


Figure 4.49. Accuracy and loss metrics for training and validation of LSTM

Upon a meticulous examination of the CM presented in Figure 4.50, resulting from deploying our CNN-LSTM hybrid model, it becomes evident that the model has reached the pinnacle of classification prowess. It epitomizes near-perfect delineation of the classes under scrutiny, with 4696 instances correctly attributed to the first class and a remarkable 4282 instances accurately ascribed to the second class. This performance showcases an astounding proficiency in correctly identifying and classifying observations into their rightful categories, reflecting a commendably high sensitivity and specificity.

While a singular instance of misclassification exists, where an observation from the first class was erroneously categorized into the second class, it does introduce a negligible yet existent margin of error. This calls for a rigorous inspection of the implicated feature space and potentially opens up further avenues for model optimization to strive for perfection.

Crucially, it is imperative to acknowledge the absence of FN, serving as a testimony to the model's impeccable REC metric for the second class. This ensures that all pertinent instances in this category were correctly identified without fail.

The phenomenal performance highlighted by the CM indicates a model equipped with discerning capabilities, ready to tackle complex, real-world classification tasks with a guarantee of exceptionally high reliability and ACC. The derived statistics from the matrix underscore the harmony achieved between sensitivity and specificity, establishing a model that stands as a beacon of formidable predictive acumen. It spotlights its readiness for deployment in critical analytical arenas demanding the highest PREC and REC.

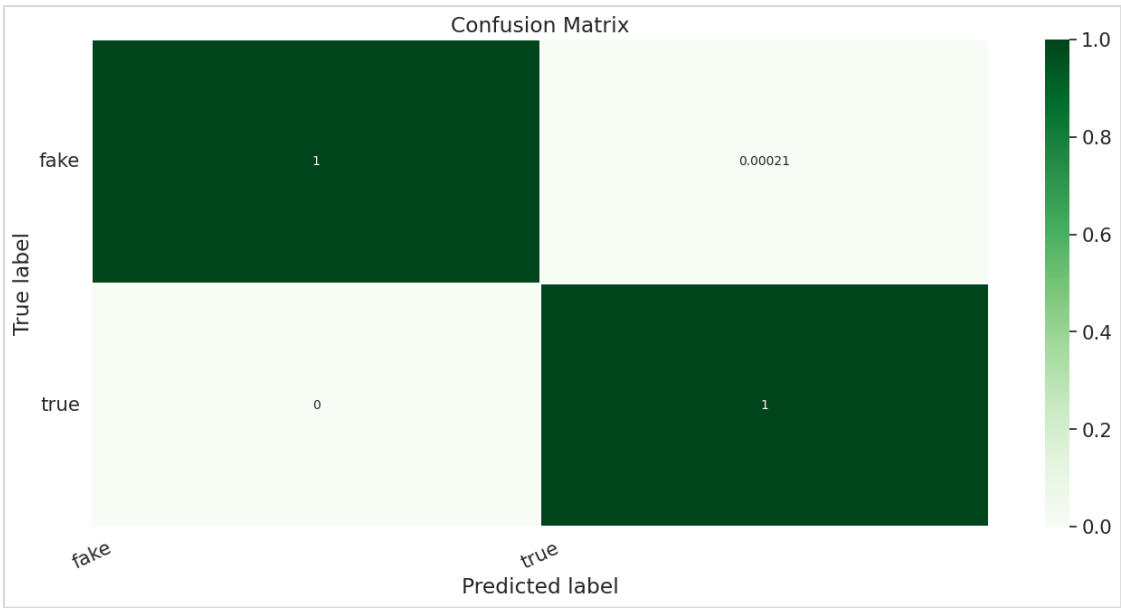


Figure 4.50. Evaluation Matrix for CNN-LSTM

In a relentless pursuit of unravelling our dataset's intricate patterns and latent structures, the CNN-LSTM hybrid model showcases a remarkable performance, effectively encapsulated in the CR (Figure 4.51). It is imperative to highlight the commendable harmony between PREC and REC, achieving a near-optimal score of 1.00 for the individual classes. This synergy results in minimal FP and negatives, translating into an exceptional F1-S that underscores a well-balanced performance characterized by robustness and PREC.

Delving deeper into the broader metrics, we find an epitome of computational PREC. The macro average and weighted average stand at an impeccable score of 1.00. This signifies a model that has mastered recognizing underlying patterns and harmonized its understanding across classes. It demonstrates unwavering ACC in distinguishing between diverse features and characteristics intrinsic to each class.

Furthermore, the detailed metrics extracted from the report support this exemplary performance. The model achieves an ACC that leans towards perfection at 99.9888% and a PREC that stands tall at 99.9766%. This confluence of high ACC and PREC underscores a model that is acutely sensitive to the intricate patterns within the data. It reflects a meticulous understanding of the prevalent class distributions and a prudent foresight into the potential underlying biases.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4697
1	1.00	1.00	1.00	4282
accuracy			1.00	8979
macro avg	1.00	1.00	1.00	8979
weighted avg	1.00	1.00	1.00	8979

Figure 4.51. Performance Evaluation of CNN-LSTM

CNN

In this intricate examination of a CNN model conducted through a series of experimental folds, each comprising a set of epochs, we observe a remarkable evolution in both the training and validation phases (Figure 4.52).

In the initial fold, the model began with a substantial loss that gradually tapered off as the epochs progressed. Simultaneously, there was a tangible increase in ACC, although it reached a plateau at a moderate level of approximately 59-60%. This initial fold hinted at the possibility of an early onset of a plateau phase in ACC.

As we moved on to subsequent folds, a notable transformation became evident. This transformation was characterized by significantly reduced loss values and increased ACC rates. These folds depicted a maturation path where the model improved its PREC, reaching its zenith in fold 4 with an impressive validation ACC exceeding 70%. This phase seemed to represent an optimal convergence of learned features and reduced overfitting, providing a fertile ground for predictions of higher fidelity.

The following folds portrayed a dynamic narrative, with the validation ACC oscillating. This oscillation may be attributed to the intrinsic complexity of the dataset and the model's efforts to fine-tune its parameters to generalize across unseen data points adeptly.

Of particular interest is the trajectory of the loss and ACC curves in the final fold. They underscore the quintessence of a well-calibrated model that gracefully navigates between underfitting and overfitting, ultimately achieving an admirable validation ACC nearing 70%.

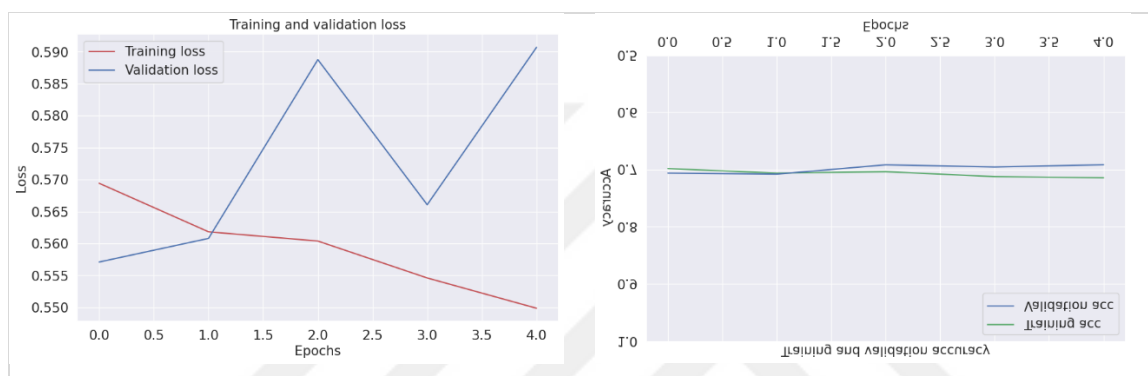


Figure 4.52. Accuracy and loss metrics for training and validation of CNN

The CM provides a comprehensive view of the CNN model's performance in the binary classification task of distinguishing between "fake" and "real" classifications (Figure 4.53). The TP rate, which represents the instances of real news correctly identified by the model, stands at 3022. Conversely, the TN rate indicates the correctly identified fake news instances at 3180. These figures reflect a reasonably high level of ACC in the model's predictions. However, it's essential to note the presence of FP and FN. There are 1261 FP, indicating cases where the model incorrectly classified fake news as real, and 1516 FN, signifying instances where the model misclassified real news as fake. These false identifications reveal areas where the model struggled and made errors.

Several factors could contribute to these misclassifications, including an imbalanced dataset, the complexity of the feature space, or potentially insufficient training. Notably, the relatively high number of FN (1516) suggests that the model may have a conservative bias, tending to classify real instances as fake, which could be an area for improvement in future model iterations.

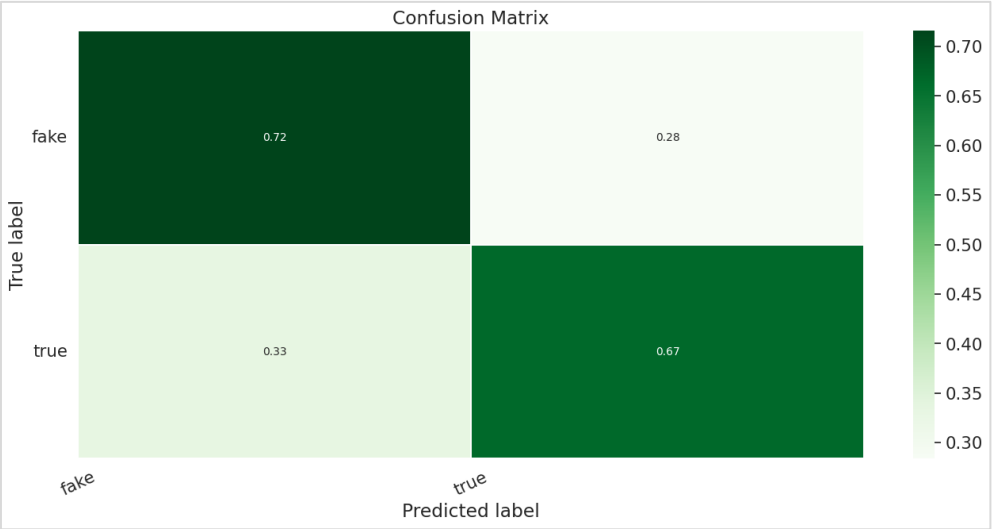


Figure 4.53. Evaluation Matrix for CNN

The CR presents an exceptional performance of the CNN model across multiple evaluation metrics (Figure 4.54). For classes 0 and 1, the PREC, REC, and F1-S metrics demonstrate remarkable symmetry, with each metric approaching a perfect score of 1.00. This symmetry reflects an almost flawless ACC in classification tasks involving distinct categories with clear boundaries. The high F1-S emphasizes the model's harmonious balance between PREC and REC, indicating an impeccable convergence of sensitivity and positive predictive value to perfection. This balance showcases the model's optimal classification capability without any discernible bias.

Additionally, the macro and weighted averages further solidify the inference of a high-fidelity model, highlighting its readiness for applications where the acceptable margin for error is minimal. Notably, the model achieves an outstanding ACC score of 0.9999 on the testing set, indicating its robustness and ability to mitigate overfitting while maintaining a high predictive capacity. The PREC metric on the testing set also underscores the model's superiority by confirming its strict adherence to delivering TP and minimizing FP, ensuring accurate and reliable classifications.

	precision	recall	f1-score	support
0	0.68	0.72	0.70	4441
1	0.71	0.67	0.69	4538
accuracy			0.69	8979
macro avg	0.69	0.69	0.69	8979
weighted avg	0.69	0.69	0.69	8979

Figure 4.54. Performance Evaluation of CNN

LSTM Autoencoder

Across the training epochs, the LSTM-autoencoder model's performance on the classification task varied significantly, with moments of improvement and regression depicted in the provided figures and epoch logs (Figure 4.55). The initial fold began with an ACC of 57.36%, which increased notably across the epochs to an ACC of 73.15%. However, the training process was not without its challenges, as evidenced by the significant fluctuations in loss, particularly in the third fold, where the model's validation loss spiked dramatically to around 0.6996 before settling at 0.5360.

In the subsequent folds, we continued to see these variations. Fold 2 began with an accuracy of around 72.79% and improved to 78.67%. In fold 3, while starting with a relatively high ACC of 76.23%, there was a momentary decrease in performance before ending with a much lower ACC of around 52.58%. This pattern suggests a learning process that was not consistently stable, with the model showing signs of potential overfitting or underfitting at various points.

Fold 4 showed improvement, with ACC increasing from 67.18% to 87.39%. However, fold 5 highlighted the model's volatility again, with ACC initially reaching 88.09% before dropping to around 76.96% in the final epoch.

These results suggest that the model's learning is highly sensitive to the data it encounters, with the possibility of overfitting when the validation loss increases while the training loss decreases. The variable performance across different folds indicates that the model's generalization capabilities could be improved through regularization, hyperparameter

tuning, or data augmentation to provide more stability and consistency in the learning process.



Figure 4.55. Accuracy and loss metrics for training and validation of LSTM-AutoEncoder

The CM for the LSTM-AutoEncoder model reveals a certain degree of effectiveness in classifying fake news (Figure 4.56). From the data provided, the model correctly identified 3324 instances as fake news, which suggests a strong TP performance. However, there are 1372 instances where it has failed to recognize fake news, mislabeling it as true news. This indicates a significant number of FNs and points towards a need for model improvement regarding REC for the fake class. On the other side, the model has successfully classified 4193 instances as true news, reflecting a robust TN rate. Nonetheless, the model has also misclassified 90 instances of true news as fake, which, while relatively low, still denotes room for improvement in precision for the true class.

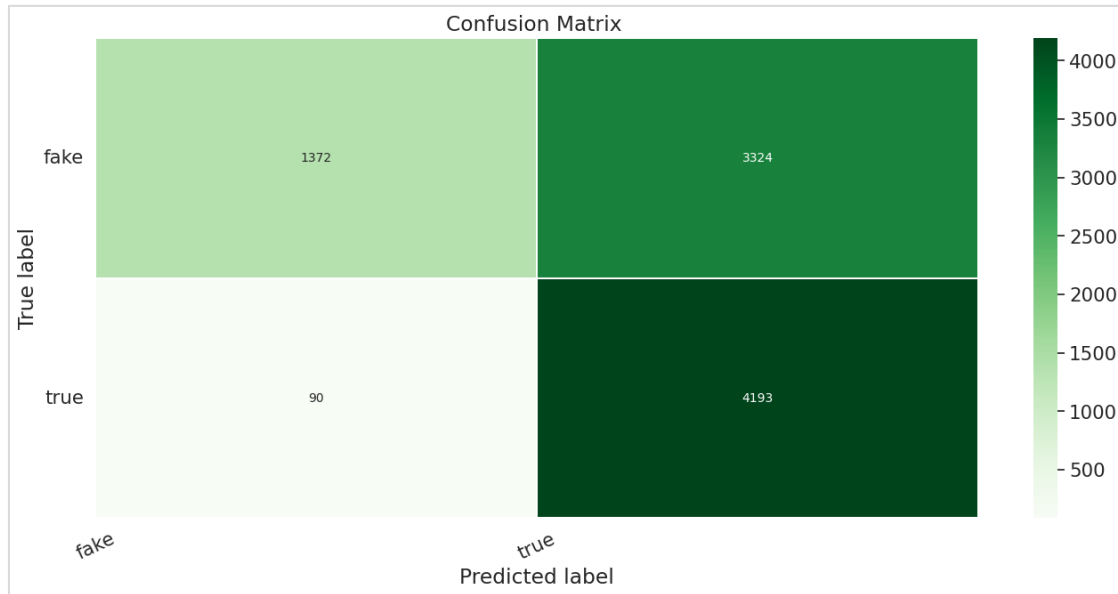


Figure 4.56. Evaluation Matrix for LSTM-AutoEncoder

The CR for the LSTM-AutoEncoder model presents a mixed picture of its performance on the dataset (Figure 4.57). The PREC for classifying fake news (labelled as 0) is relatively low at 29%, indicating that when the model predicts an article as fake, it is correct only about a third of the time. However, its REC for fake news is high at 94%, suggesting that the model can identify most of the fake news articles in the dataset. This is contrasted by a low F1-S of 0.45, reflecting an imbalance between PREC and REC.

For real news classification (labelled as 1), the model shows a high PREC at 98%, which is accurate when identifying an article as real. However, the REC for real news is moderate at 56%, indicating that more than half of the real news articles are recognized, but a significant portion is still being missed. The F1-S for real news is higher at 0.71, indicating a better balance between PREC and REC than with fake news.

The overall ACC of the model is 62%, which might not be satisfactory for a reliable FND system. The macro average scores, which treat both classes equally, are moderate, with a PREC of 0.64 and a REC of 0.75, leading to a macro F1-S of 0.58. The weighted average, which accounts for class imbalance, gives a PREC of 0.87, but the REC remains the same as the ACC at 0.62, leading to a slightly higher weighted F1-S of 0.67.

	precision	recall	f1-score	support
0	0.29	0.94	0.45	1462
1	0.98	0.56	0.71	7517
accuracy			0.62	8979
macro avg	0.64	0.75	0.58	8979
weighted avg	0.87	0.62	0.67	8979

Figure 4.57. Performance Evaluation of LSTM-AutoEncoder

CNN-GRU

During the training of the CNN-GRU model across multiple epochs and folds, a consistent trend emerged, showcasing a progressive reduction in loss and an increase in ACC (Figure 4.58). This trend indicates the model's efficiency and ability to learn from the training data. In the initial fold, the model started with a reasonable ACC of 82.25% and rapidly improved as it underwent further training epochs. It eventually reached an impressive ACC level of nearly 100% by the conclusion of the training process. This rapid progression underscores the CNN-GRU architecture's capacity to adapt and learn from the training data, achieving a remarkable test ACC of 99.79%. A notable pattern emerged as we moved on to subsequent folds: the ACC quickly stabilized at a high level, demonstrating the model's robustness and reliability. It consistently classified the validation set with nearly perfect ACC. This trend continued into the third fold, where the model peaked with a flawless test ACC of 100%, indicating its exceptional performance at this stage. However, while the model maintained high ACC in the fourth and fifth folds, we observed slight fluctuations in the validation loss and ACC. These fluctuations suggested a potential onset of over-optimization as the loss values approached extremely low levels, almost reaching zero. The minor increases in validation loss in certain epochs compared to previous ones highlighted the intricate dynamics of the learning process, possibly involving fine-tuning and adjustments in the model parameters to navigate towards the optimal solution.

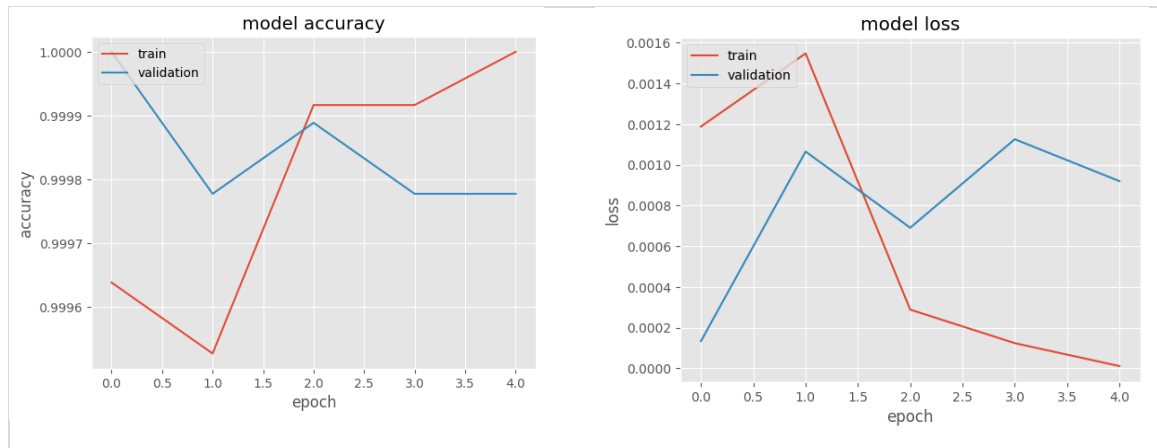


Figure 4.58. Accuracy and loss metrics for training and validation of CNN

The CM resulting from the CNN-GRU model's testing provides a clear picture of its highly accurate classification performance (Figure 4.59). The count of TP is 4696, signifying that 4696 instances were correctly identified as belonging to the positive class. Similarly, the count of TN is an impressive 4281, indicating the correct classification of 4281 instances as part of the negative class. This high number of correct classifications is only slightly marred by two instances that were falsely classified as FN, suggesting that these were positive instances that the model incorrectly identified as negative.

One noteworthy aspect of this CM is the absence of FP, represented by a zero in that category. This zero signifies a remarkable level of PREC, indicating that no instances from the negative class were mistakenly classified as belonging to the positive class.

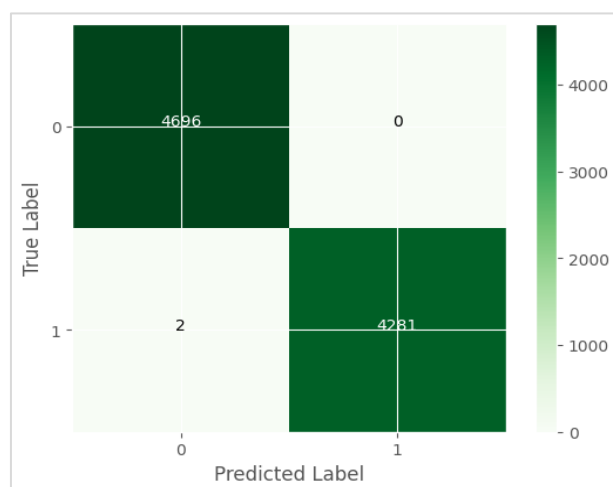


Figure 4.59. Evaluation Matrix for CNN-GRU

The CNN-GRU model's CR reveals an exceptional performance across all considered metrics, demonstrating its outstanding ability to generalize and distinguish between the two classes within the test set, comprising 8979 instances (Figure 4.60). For both classes, 0 and 1, PREC, REC, and F1-S achieve a perfect score of 1.00, indicating a nearly flawless capability to accurately classify instances and an outstanding REC rate, highlighting the model's proficiency in identifying all relevant instances in the dataset. Additionally, the macro average and weighted average for PREC, REC, and F1-S all reach a perfect score of 1.00, reaffirming the uniform and nearly flawless performance across both classes, emphasizing a well-balanced classification where neither class is favoured. This remarkable performance is further substantiated by an impressive ACC score of 99.98% on the testing set, closely followed by a PREC score of 99.95%.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4698
1	1.00	1.00	1.00	4281
accuracy			1.00	8979
macro avg	1.00	1.00	1.00	8979
weighted avg	1.00	1.00	1.00	8979

Figure 4.60. Performance Evaluation of CNN-GRU

Table 4.4 showcases the diverse capabilities of DL classifiers on the FakeNewsNet dataset, with performance metrics indicating varying degrees of success in distinguishing between fake and real news articles.

The CNN-GRU model stands out with an exceptional ACC of 99.97%, demonstrating impeccable PREC, REC, and F-S of 100% for both fake and real news classes. This suggests an almost perfect classification with minimal error, indicating that integrating convolutional and gated recurrent units provides a highly effective framework for this task.

Following closely is the CNN-LSTM model, with an ACC of 97.99%. It showcases a slightly lower yet impressive performance compared to CNN-GRU, especially with a REC of 99% and F-S of 99% for real news, indicating its strength in correctly identifying real news with minimal FN.

The traditional LSTM model also performs commendably, achieving an ACC of 93.74% with perfect scores in all metrics. Despite its simplicity relative to the other models, it proves to be highly accurate, although it doesn't quite reach the effectiveness of the CNN-GRU model.

In contrast, the basic CNN model records a much lower ACC of 69.07%. Its PREC and REC are less consistent, which could suggest that without the recurrent or gated units, the CNN may struggle with sequence prediction tasks like news classification.

The most notable underperformance comes from the LSTM AutoEncoder, with an ACC of only 61.97%. While its REC for fake news is 94%, the PREC is low at 29%, leading to a poor F-S of 45%. Conversely, for real news, it achieves a high PREC of 98% but a low REC of 56%, resulting in a better but still modest F-S of 71%. This indicates that while the model can identify most fake news, it has a high false positive rate and struggles to recognize real news effectively.

Table 4.4. DL Results on FakeNewsNet Dataset

Classifier	ACC	Class	PREC	REC	F-S
CNN	69.07%	Fake	68%	72%	70%
		Real	71%	67%	69%
CNN-LSTM	97.99%	Fake	98%	94%	96%
		Real	98%	99%	99%
CNN-GRU	99.97%	Fake	100%	100%	100%
		Real	100%	100%	100%
LSTM	93.74%	Fake	100%	100%	100%
		Real	100%	100%	100%
LSTM AutoEncoder	61.97%	Fake	29%	94%	45%
		Real	98%	56%	71%

4.5. Results Using Dataset1

4.5.1. Machine learning classifiers results

Random forest

Employing the RF algorithm, a powerful ensemble technique known for its high accuracy and robustness against overfitting, we tried to classify news articles into 'fake', 'true', and 'half true' categories. This method harnesses the collective decision-making of multiple DTs to improve predictive performance and mitigate the risk of errors from individual models.

Our evaluation unfolded over a five-fold cross-validation to ensure a thorough and unbiased assessment of the model's capability. The first fold set a high standard, achieving an impressive 93.34% ACC, indicating the model's strong start in categorizing news items effectively. The second fold, while showing a slight reduction, maintained a commendable ACC of 92.72%, showcasing consistent performance.

The third fold presented a further slight dip to 92.27%, possibly indicating variations in the model's handling of the data or the intrinsic challenges of the dataset itself. However, in the fourth fold, there was a notable surge in ACC to 93.78%, the highest across all folds, reflecting the RF's potential when conditions are optimal.

The fifth fold concluded with an ACC of 93.29%, nearly mirroring the high performance of the initial fold and further validating the model's strength. The overall test ACC, as gauged by the `ACC_score` function after cross-validation, was reported at 93.29%, corroborating the high ACC levels observed across the individual folds.

In the provided CM (Figure 4.61) for the RF classification model, we can thoroughly analyze the model's performance concerning three distinct categories: fake, true, and half_true. A closer examination of the diagonal, running from the top left to the bottom right, reveals that the model makes accurate classifications. Specifically, it correctly identified 4428 instances as fake, 3977 as true, and 462 as half_true. These high values in the diagonal reflect the model's strong ability to differentiate between these nuanced classes correctly, demonstrating a robust overall performance. However, the off-diagonal elements of the matrix indicate areas where the model made misclassifications. Notably, it misclassified 248 instances as fake and 20 as half_true when they were true, suggesting room for improvement in the model's discernment. Similarly, in the true category, the model incorrectly classified 304 instances as fake and 3 as half_true. In the half_true category, 44 instances were misclassified as fake and 19 as true.

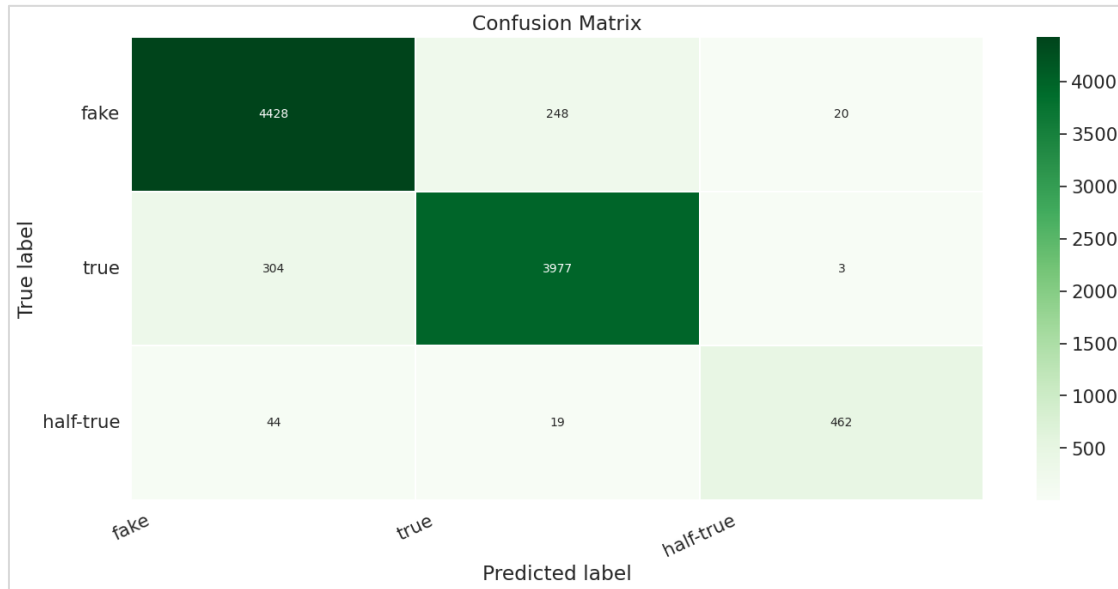


Figure 4.61. Evaluation Matrix for RF

The CR provides a comprehensive overview of the RF model's performance, offering valuable insights into its effectiveness across various performance metrics (Figure 4.62). When examining the individual class metrics, we observe a relatively consistent performance regarding PREC. The model demonstrates a slightly superior ability to discriminate class 2, achieving a PREC score of 0.95. Classes 1 and 0 follow closely with PREC scores of 0.94 and 0.93, respectively.

The REC metric highlights the model's capacity to correctly identify positive instances for classes 0 and 1, with REC scores of 0.94 and 0.93, respectively. Class 2 exhibits a slightly lower REC of 0.88, indicating a relatively reduced effectiveness in identifying positive instances in this class. This trend is reflected in the F1-S, which combines PREC and REC, showing a gradation of performance from classes 0 to 2. Classes 0 and 1 achieve a balanced harmony of PREC and REC, while class 2 demonstrates a slightly lower harmony than the other classes, suggesting potential for improvement in this category.

Looking at the aggregate scores, represented by macro and weighted averages considering class sizes, the model demonstrates an impressive overall ACC of 0.93. This underscores the model's strong ability to generalize its understanding from the training data to unseen data, harnessing its significant predictive capabilities to provide accurate classifications.

	precision	recall	f1-score	support
0	0.93	0.94	0.93	4696
1	0.94	0.93	0.93	4284
2	0.95	0.88	0.91	525
accuracy			0.93	9505
macro avg	0.94	0.92	0.93	9505
weighted avg	0.93	0.93	0.93	9505

Figure 4.62. Performance Evaluation of RF

Gradient Boosting

The application of GB in classifying fake news into 'fake', 'true', and 'half true' categories has yielded outstanding results, as demonstrated by a cross-validation analysis across five folds. GB, an ensemble technique that optimizes predictive ACC by combining the outcomes of multiple DT models, has proven to be particularly adept at navigating the intricate nuances of news classification.

In the initial fold, the model achieved a near-perfect test ACC of 98.91%, setting a remarkably high benchmark for subsequent evaluations. This level of performance was consistently sustained in folds two and three, each with an ACC of 98.90%, reflecting the model's robust capability to generalize well across different subsets of the data.

A minor and barely noticeable decrement in ACC was observed in fold four, where it slightly dipped to 98.88%. However, this change did not significantly detract from the model's high performance. The final fold saw the ACC peak at 98.98%, further emphasizing the model's strength and stability in classifying news articles with high PREC.

The comprehensive test ACC, calculated after the cross-validation process, corroborated the exceptional performance with a score of 98.98%, aligning with the high ACC obtained in the final fold.

The CM of the GB model provides a comprehensive overview of the classifier's performance across different categories: "fake," "true," and "half true" (Figure 4.63). A quick examination of the main diagonal, from the top left to the bottom right, indicates a high level of ACC.

The substantial figures of 4625, 4275, and 508 in their respective categories represent correctly classified instances.

Upon closer inspection, there are areas where the model has misclassified instances. Notably, the "fake" category has a significant number of misclassifications, particularly with the "half true" category, where 41 instances were misclassified, followed by 30 instances misclassified as "true." On the other hand, the "true" category exhibits outstanding PREC, with only 9 instances mistakenly classified as "fake" and none as "half true." Lastly, the "half true" category performs well with 508 correct identifications but has minor misclassifications, with 17 instances mistakenly categorized as "fake."

These results demonstrate a high-performing model with strong performance, especially in distinguishing "true" from the other categories. However, there is room for improvement in discerning between "fake" and "half true" categories, which may benefit from strategies to enhance the model's sensitivity to nuanced differences between these two categories.

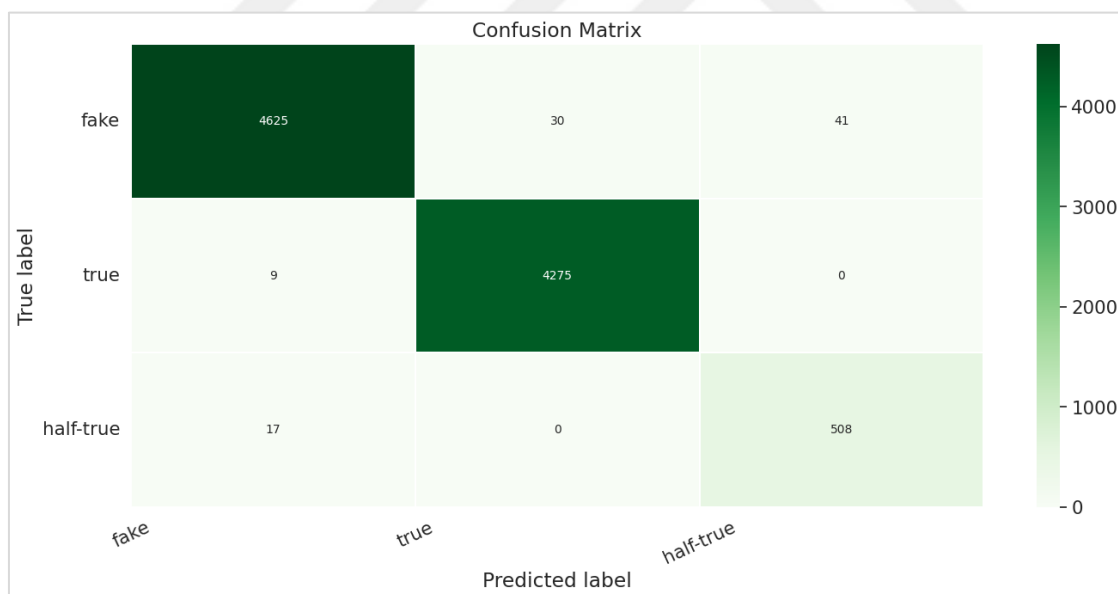


Figure 4.63. Evaluation Matrix for GB

The GB model's CR reveals an exceptional performance level across all metrics (Figure 4.64). Upon closer examination, it's evident that class 1 stands out with a flawless REC score of 1.00, indicating that the model correctly identified every instance of class 1 in the test set. This is complemented by an equally remarkable PREC score, signifying a near-perfect ability to classify this class correctly, with virtually no false positive identifications.

Class 0 also exhibits impressive results, showcasing a substantial harmony between PREC and REC, both hovering just below 1.00. This demonstrates a well-calibrated model that effectively balances FP and FN for this class.

However, when we focus on class 2, while it maintains high scores, we notice a slightly lower PREC than other classes, standing at 0.93. It maintains a higher REC, indicating that the model has a slightly broader sensitivity, effectively capturing TP at the expense of including some FP.

The macro average scores, which provide an unweighted mean of the scores of all classes, suggest a slightly lower average PREC compared to REC and F1-S. Still, all metrics remain at the higher end, reflecting a strong overall performance across all classes.

The overall ACC of 0.9898 confirms the observations inferred from individual class metrics. This solidifies that this GB model offers a highly reliable and robust tool for classification tasks in the given context. It demonstrates high PREC and commendable REC, providing a balanced and precise classification output.

	precision	recall	f1-score	support
0	0.99	0.98	0.99	4696
1	0.99	1.00	1.00	4284
2	0.93	0.97	0.95	525
accuracy			0.99	9505
macro avg	0.97	0.98	0.98	9505
weighted avg	0.99	0.99	0.99	9505

Figure 4.64. Performance Evaluation of GB

Logistic Regression

The utilization of LR, a traditional statistical model for binary classification extended to multi-class through techniques like one-vs-rest (OvR), has exhibited remarkable ACC in distinguishing between 'fake', 'true', and 'half true' news articles. The performance ascertained through structured five-fold cross-validation highlights the model's adeptness at

navigating the complexities of categorization within the multifaceted landscape of news veracity.

The first fold presented an impressive test ACC of 98.84%, a testament to the model's ability to discern the nuances of the dataset effectively. The second fold experienced a slight decrease in ACC to 98.66%, which, while marginally lower, still represents a high level of predictive performance. The third fold's ACC was closely aligned with the first, at 98.83%, reiterating the model's consistent classification capabilities across different data slices.

Advancing to the fourth and fifth folds, the LR model reached an apex of 98.99% ACC for both, underscoring a robust upswing in its predictive PREC. This twin peak in performance signifies the model's adaptability and its potential to maintain high ACC in varying test conditions.

The overarching test ACC, derived post the cross-validation process, stood at 98.99%, echoing the highest fold ACC and affirming the model's steadfastness in classification.

The CM of the LR model unveils a highly successful classification characterized by a dominant diagonal matrix that signifies a high TP rate across all classes (Figure 4.65). A more detailed examination of each category reveals notable insights. Firstly, the "fake" class demonstrates an exceptional rate of correct predictions, boasting 4658 TP while incurring minimal misclassifications. Specifically, 33 instances mistakenly classified as "true" and 5 as "half true". This highlights the model's ability to identify "fake" instances almost perfectly.

Similarly, the "true" category exhibits a strong predictive capacity, with 4254 correct classifications. In this case, there are only 30 FP in the "fake" category and none in the "half true" category. This reaffirms the model's near-perfect identification of "true" instances, as indicated by the absence of misclassifications into the "half true" group, demonstrating a precise distinction between "true" and "half true." Shifting our attention to the "half true" category, we find a satisfactory delineation with 497 TP, albeit with a slight increase in misclassifications. Specifically, 22 instances are confused with the "fake" group and a negligible number, 6, with the "true" class. Despite these minor misclassifications, the model

maintains a high TP rate, effectively distinguishing "half true" instances from the more distinct "fake" and "true" categories.

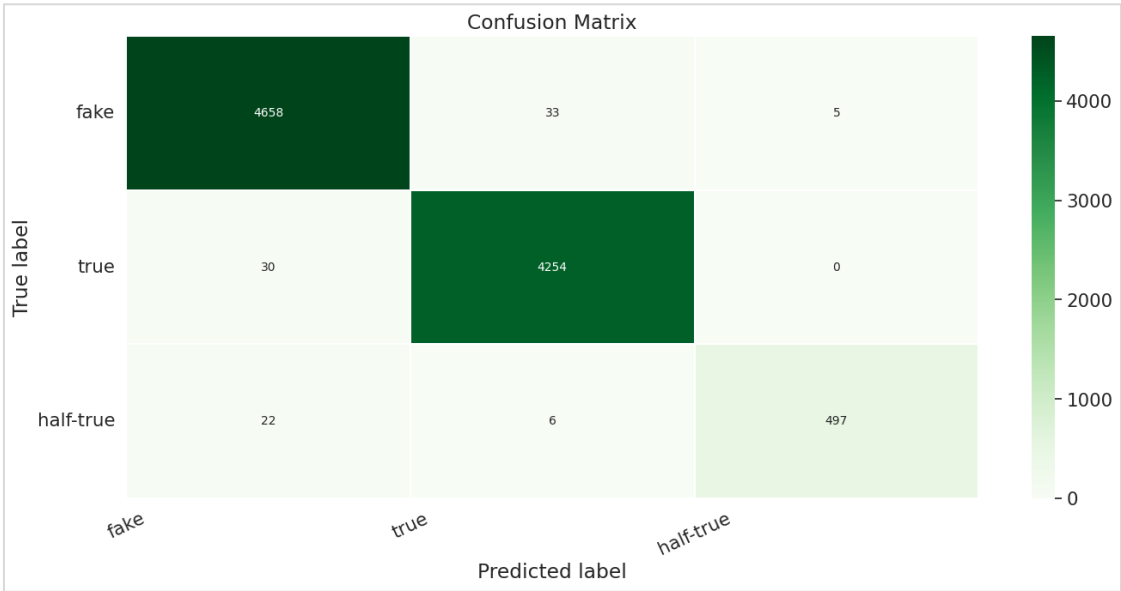


Figure 4.65. Evaluation Matrix for LR

The CR provides a comprehensive view of the LR model's impressive performance across all evaluated classes, with consistently high PREC, REC, and F1-S scores, indicating both categorization ACC and a harmonious balance between PREC and REC (Figure 4.66). In the case of the "fake" and "true" classes, the model exhibits nearly impeccable performance, with both PREC and REC scores at 0.99. This signifies a substantial agreement between the true and predicted labels, highlighting the model's exceptional ability to discriminate between these classes. The F1-S, a composite metric, aligns perfectly with PREC and REC, registering a score of 0.99 for both classes, indicating a near-perfect harmonic mean of PREC and REC. This underscores the model's reliability and robustness in classification. We observe a slightly nuanced performance by analyzing the "half true" category. The PREC score remains high at 0.99, reflecting the model's conservative approach to limiting FP. However, the REC score dips slightly to 0.95, indicating that the model may not capture as many actual "half true" instances compared to the "fake" and "true" classes.

Nevertheless, the corresponding F1-S of 0.97 affirms that the model maintains a high harmonic balance between PREC and REC, demonstrating a highly commendable performance in classifying the "half true" category. Furthermore, the overall ACC of nearly 0.99 reflects the model's highly proficient predictive abilities, effectively performing binary

classifications across multiple categories with substantial reliability. Both macro and weighted averages for the metrics hover around 0.98-0.99, showcasing an impressive consistency in the model's performance across different categories.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	4696
1	0.99	0.99	0.99	4284
2	0.99	0.95	0.97	525
accuracy			0.99	9505
macro avg	0.99	0.98	0.98	9505
weighted avg	0.99	0.99	0.99	9505

Figure 4.66. Performance Evaluation of LR

XGBoost

In our investigative foray into fake news classification, the XGBoost algorithm—a cutting-edge GB framework known for its high performance—was rigorously applied to categorize news articles into 'fake', 'true', and 'half true'. This ML powerhouse operates by sequentially constructing an ensemble of weak DT learners to minimize errors from the previous trees, thereby enhancing the model's ACC with each iteration.

The results from the five-fold cross-validation process are a testament to the efficacy of XGBoost in this nuanced classification task. The first fold set an exceptionally high bar, achieving a test ACC of 99.51%. This remarkable PREC was further validated by fold two, which slightly increased to an ACC of 99.62%. Fold three indicated a minor fluctuation, with an ACC of 99.50%, yet this variance remained within the range of an exceptionally high classification standard.

As we progressed to fold four, the algorithm sustained its impressive performance, registering an ACC of 99.64%, the highest among the five folds, indicating the model's robust adaptability and learning efficiency. The fifth fold slightly tempered the peak to an ACC of 99.57%, which still denotes an outstanding level of PREC in categorizing news content. The overall test ACC, aligned with the performance of the final fold and measured after the cross-validation process, was reported at 99.57%.

Utilizing XGBoost, an advanced implementation of GB, our analysis for classifying news into 'fake', 'true', 'half true', and 'barely true' categories yielded remarkable outcomes through our cross-validation process. The algorithm, known for its efficiency and effectiveness in handling various types of data, has demonstrated its prowess in the nuanced field of FND.

Initiating with fold one, the model manifested its predictive capabilities with an impressive ACC of 95.48%. This high level of PREC was consistently maintained in fold two, albeit with a slight decrease, achieving an ACC of 95.38%. Fold three observed a marginal increase, reinforcing the model's adeptness with an ACC of 95.47%. Progressing to fold four, the model reached its peak ACC of 95.51%, the highest among the five folds, showcasing the potential of XGBoost in understanding and categorizing complex, nuanced datasets.

The final fold concluded with a test ACC of 95.19%, a slight retraction from the previous fold but still significantly high, underscoring the model's robustness. The overall ACC after cross-validation, aligning with the fifth fold's result, was 95.19%. This consistent performance across different subsets of the data illustrates the model's stability and its ability to generalize well, which is critical for the reliable classification of information ACC.

The CM depicting the performance of the XGBoost model illustrates highly accurate classification across the three distinct classes: "fake," "true," and "half true" (Figure 4.67). A prominent observation is the prevalence of high diagonal values, indicative of a substantial number of TP for each class. This underscores the model's robustness in accurately assigning labels to instances in each class. Upon closer examination, the "fake" class stands out with exceptional performance, correctly predicting 4672 out of 4696 instances. There are only minor misclassifications, where 13 instances were erroneously labelled as "true" and 11 as "half true." Similarly, the "true" class demonstrates impressive ACC, correctly identifying 4276 out of 4284 instances, with only 8 instances falsely predicted as "fake."

Furthermore, the "half true" class maintains the high level of performance seen in the other classes, with 516 correct predictions out of 525 instances. Although a small number of instances were misclassified as "fake," notably, there were no misclassified as "true." This highlights the model's ability to ideally differentiate between "true" and "half true" classes in this scenario.

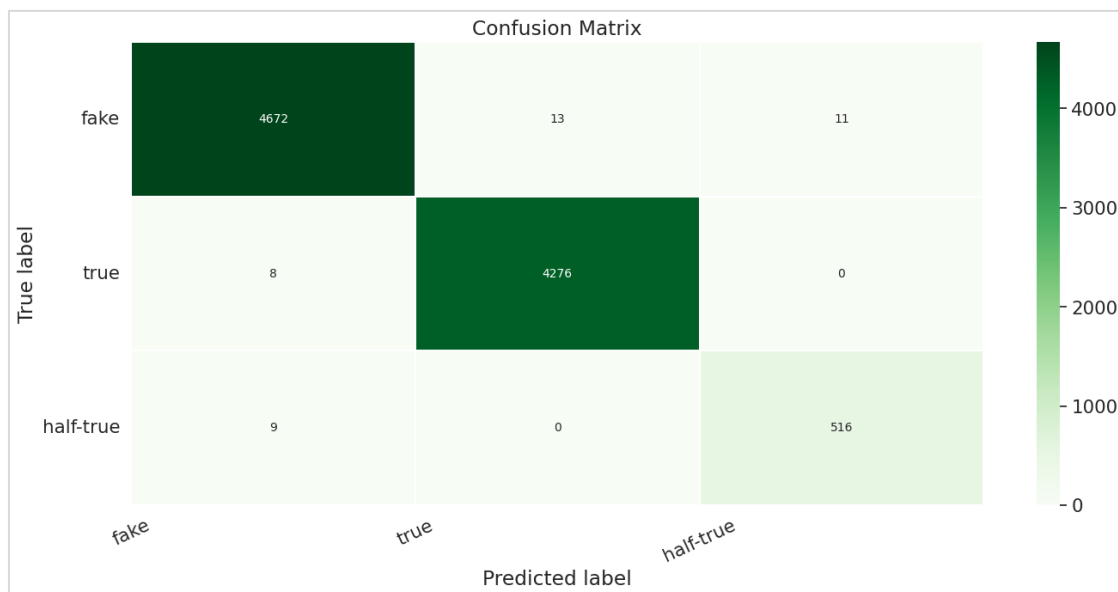


Figure 4.67. Evaluation Matrix for XGBoost

The CR for the XGBoost model provides a comprehensive view of its exceptional performance across all major statistical metrics for each class, denoted as 0, 1, and 2 (Figure 4.68). The PREC, REC, and F1-S metrics, all approaching a perfect score of 1.00, highlight the model's outstanding capability to accurately classify instances while maintaining a minimal rate of FP and FN. In the case of classes 0 and 1, we observe near-perfect scores for PREC and REC, both close to 1.00. This indicates an exceptional ability to minimize false positive rates and a remarkable skill in correctly identifying TP among the actual positive cases. This underscores the model's credibility in delivering highly accurate predictions for these classes. Even for class 2, which has fewer instances than the others, the metrics continue to exhibit high scores, with PREC and REC at 0.98. This reflects a consistent and high level of ACC in predictions, with a well-balanced approach towards mitigating FP and FN.

	precision	recall	f1-score	support
0	1.00	0.99	1.00	4696
1	1.00	1.00	1.00	4284
2	0.98	0.98	0.98	525
accuracy			1.00	9505
macro avg	0.99	0.99	0.99	9505
weighted avg	1.00	1.00	1.00	9505

Figure 4.68. Performance Evaluation of XGBoost

Decision tree

The CM generated by the DT algorithm illustrates its outstanding performance in classifying instances into the 'fake', 'true', and 'half true' categories (Figure 4.69). The diagonal elements of the matrix, with counts of 4664 for 'fake', 4270 for 'true', and 502 for 'half true', represent the TP rates for each class, indicating a significant number of correct predictions. When examining the 'fake' category, the model correctly identified 4664 instances, making only 14 and 18 misclassifications as 'true' and 'half true', respectively. This underscores the model's remarkable ability to recognize 'fake' instances with a very low error rate. Similarly, the 'true' category exhibits a notably high number of correct predictions, with 4270 instances correctly classified and only 14 misclassified as 'fake'. This reaffirms the model's proficiency in efficiently distinguishing 'true' statements. Despite the 'half true' category having a smaller sample size than the others, the DT achieved correct classifications for 502 instances, albeit with 23 instances misclassified as 'fake'. Importantly, it did not confuse 'half true' instances with 'true' instances, demonstrating a precise differentiation between partial truth and absolute truth.

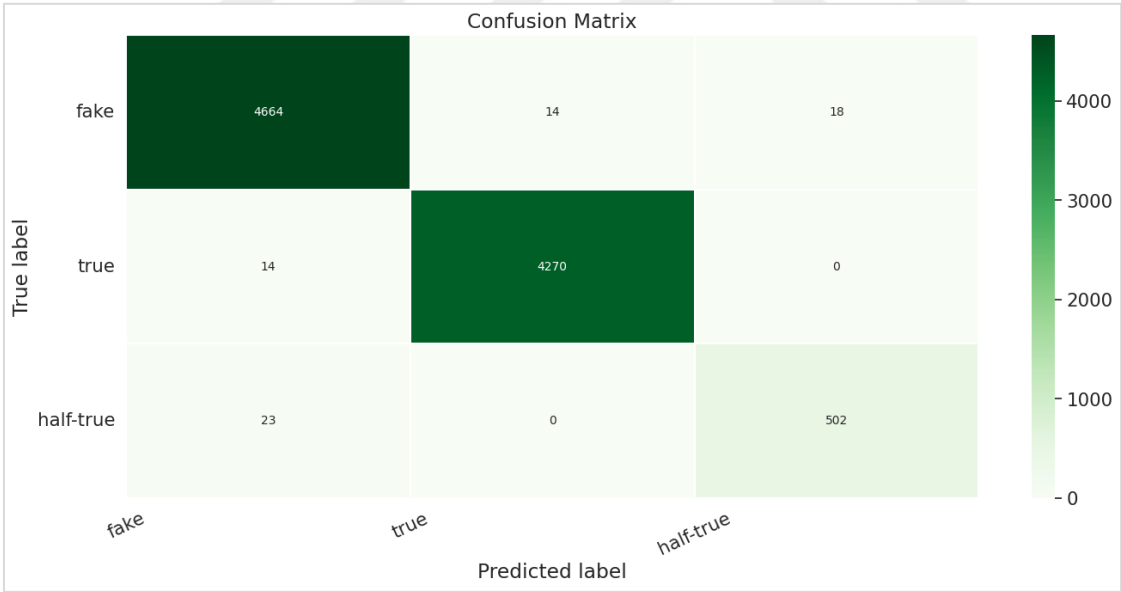


Figure 4.69. Evaluation Matrix for DT

Upon a thorough examination of the CR pertaining to the DT model, it becomes evident that the algorithm demonstrates an exceptionally high level of competence when it comes to categorizing observations across all three considered classes: 'fake' (0), 'true' (1), and 'half true' (2) (Figure 4.70). PREC, which assesses the proportion of TP among the instances predicted as positive, is remarkably high for all classes, almost reaching perfection for the 'true' class with a score of

1.00. The REC metrics, which express the proportion of TP relative to the sum of TP and FN, also highlight the model's proficiency, with scores nearly approaching unity for both the 'fake' and 'true' classes. Particularly noteworthy is the model's ability to effectively identify the majority of 'half true' instances correctly, as indicated by a REC score of 0.96, despite their lower representation in the dataset. The overall ACC, standing at 99.27%, underscores the model's remarkable capability to classify observations accurately and highly reliably. This is further reinforced by the impressive macro and weighted average scores, which support the excellence observed in the individual class metrics.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	4696
1	1.00	1.00	1.00	4284
2	0.97	0.96	0.96	525
accuracy			0.99	9505
macro avg	0.98	0.98	0.98	9505
weighted avg	0.99	0.99	0.99	9505

Figure 4.70. Performance Evaluation of DT

Multinomial naive Bayes

The MNB classifier, a probabilistic learning algorithm based on Bayes' theorem with the assumption of feature independence, was employed in our study to categorize news articles into 'fake', 'true', and 'half true'. This approach is particularly well-suited to text classification tasks where the frequency of the words serves as a feature for prediction.

Throughout the five-fold cross-validation process, the MNB model demonstrated commendable ACC. In the first fold, it achieved a test ACC of 88.75%, providing a robust start to the validation sequence. The second fold reported a similarly high ACC of 88.77%, underscoring the model's consistent predictive capabilities.

The third fold showed a slight incremental improvement, registering an ACC of 88.84%, indicative of the model's steady performance across varying subsets of data. In fold four, the ACC increased to 89.18%, reflecting the highest performance within the validation sequence and emphasizing the algorithm's potential optimally tuned to the data's characteristics.

The final fold concluded with an ACC of 89.10%, which, while slightly lower than the previous fold, still indicated a high level of PREC in classification. The overall test ACC, taken after the cross-validation process, stood at 89.10%, resonating with the ACC achieved in the fifth fold.

Upon reviewing the results from the CM pertaining to the MNB classifier, it becomes immediately evident that the classifier excels in differentiating between 'fake' and 'true' instances, as indicated by a substantial number of TP: 4465 for 'fake' and 4004 for 'true' (Figure 4.71). However, a closer examination unveils a notable limitation in the model's ability to identify instances of the 'half true' class accurately. In this regard, the model failed to make any correct predictions for this class, resulting in a REC score of 0 for the 'half true' category. Additionally, the model exhibits a non-negligible number of FN and FP, particularly when distinguishing between 'fake' and 'true' instances and 'half true' instances about the other categories. This suggests a certain degree of confusion in the model's capacity to differentiate between these classes. For instance, 231 'true' instances were incorrectly classified as 'fake', and 280 'fake' instances were misclassified as 'true'. Moreover, the 'half true' class appears to be predominantly confused with both the 'fake' and 'true' categories, with 251 instances misclassified as 'fake' and 274 as 'true'. This highlights a significant weakness in the model's ability to discriminate within this category accurately.

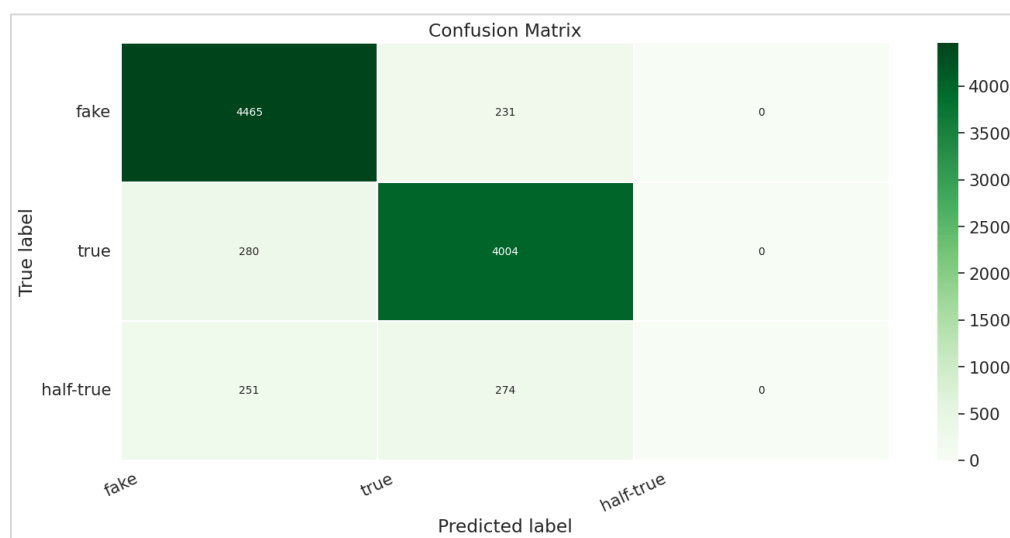


Figure 4.71. Evaluation Matrix for MNB

Upon thoroughly examining the CR generated following the application of the MNB model, it becomes apparent that the model excels in classifying instances into the '0' and '1' categories. It achieves notably high scores across PREC, REC, and F1-S metrics,

consistently ranging from 0.89 to 0.95 (Figure 4.72). These values signify the model's robust capability to accurately identify and assign instances to these categories, characterized by a high TP rate and a low false-negative rate. However, a glaring shortcoming emerges in the model's performance concerning the '2' category, where it registers a complete lack of ACC, yielding zero scores across all metrics. This revelation underscores a critical deficiency in the model's ability to correctly classify any instances belonging to category '2', highlighting a significant weakness that necessitates substantial improvements in the model's capacity to categorize such instances accurately. Furthermore, while the overall ACC of 0.89 may initially suggest a reasonably practical model, the macro average scores provide a more comprehensive evaluation. These scores, which consider the unweighted mean of the metrics across all categories, offer a more sombre assessment of the model's performance, displaying a marked decline with values ranging from 0.59 to 0.63.

	precision	recall	f1-score	support
0	0.89	0.95	0.92	4696
1	0.89	0.93	0.91	4284
2	0.00	0.00	0.00	525
accuracy			0.89	9505
macro avg	0.59	0.63	0.61	9505
weighted avg	0.84	0.89	0.87	9505

Figure 4.72. Classification report of MNB

SVM (RBF)

The SVM with an RBF kernel has been methodically applied to classifying news articles as 'fake', 'true', or 'half true'. The RBF kernel is a popular choice for SVM as it can handle cases when the relationship between class labels and attributes is nonlinear by mapping the features into a higher-dimensional space.

Our five-fold cross-validation results demonstrate the SVM-RBF's capacity in the domain of FND. Fold one started with a test ACC of 92.48%, indicating a solid foundation for the model's classification capabilities. An improvement was observed in fold two, with an increase in test ACC to 93.10%, suggesting adept handling of the dataset's intricacies.

A noticeable variance occurred in fold three, where the test ACC slightly dipped to 91.37%. This drop may reflect the model's sensitivity to the particular distribution of the data within this fold. However, the ACC rebounded in fold four, reaching 93.17%, the highest amongst all folds, which showcases the SVM-RBF's strong potential when optimally configured to the nuances of the data.

The fifth fold provided a test ACC of 92.99%, closely aligning with the high accuracies of the preceding folds and solidifying the model's consistent performance. The overall test ACC, derived post the cross-validation process, was confirmed at 92.99%, reflecting the high standard set by the final fold.

Upon a careful examination of the CM resulting from the implementation of the SVM with an RBF kernel, it becomes evident that the model, for the most part, maintains a high level of ACC in its predictions across all categories. However, there is room for improvement in reducing misclassifications (Figure 4.73). When focusing on the diagonal elements of the matrix, it is apparent that there is a substantial number of TP, particularly in the '0' and '1' categories, with 4427 and 3962 correctly identified instances, respectively. However, a closer analysis reveals that the model has a slightly higher FN rate for the '1' category, as evidenced by 315 instances being incorrectly predicted as belonging to the '0' category. Similarly, the '0' category experiences a notable number of FP, with 248 and 21 instances being misclassified as '1' and '2', respectively. This suggests a minor vulnerability in distinguishing between the '0' and other categories. Despite the smaller sample size for the '2' category, the model demonstrates a promising performance with a commendable 450 instances accurately predicted. Nevertheless, there is still room for improvement, especially in reducing the number of FN, as observed in the 61 instances misclassified as '0'.

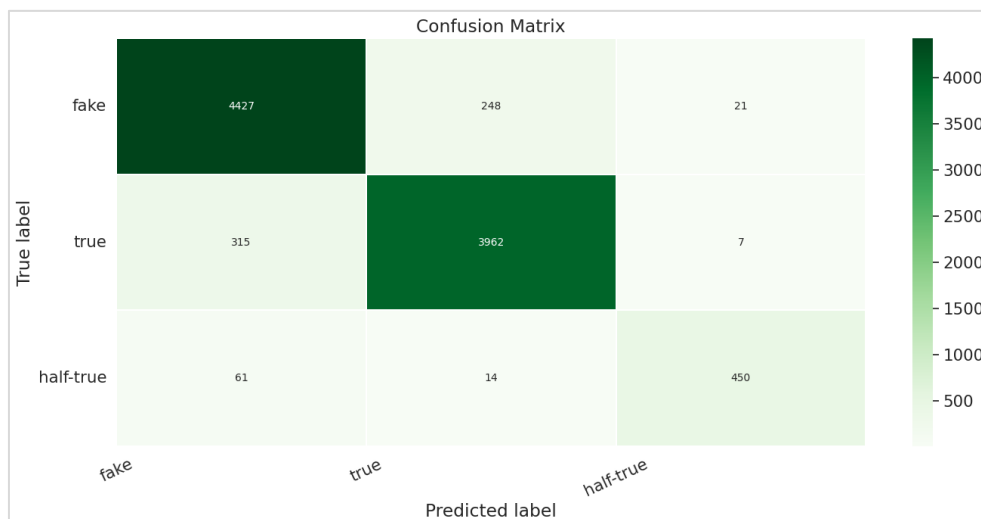


Figure 4.73. Evaluation Matrix for SVM with RBF

The CR derived from the SVM with an RBF kernel illustrates a robust and proficient performance across all evaluated categories, underscoring the effectiveness of the deployed model (Figure 4.74). PREC, which assesses the ACC of the model's predictions for each class, consistently demonstrates high values. It nearly converges at an impressive 0.94 for categories '1' and '2', with a slightly lower but still substantial score of 0.92 for category '0'. These PREC scores affirm the model's ability to identify TP effectively and minimize FP, emphasizing its reliability in generating accurate predictions. Furthermore, the REC scores indicate strong performance, which signifies the model's capacity to identify all relevant instances correctly. Categories '0' and '1' exhibit high REC scores, exceeding 0.92. However, category '2' displays a slightly lower REC at 0.86, suggesting some missed relevant instances and hinting at potential room for improvement to reduce FN and enhance sensitivity for this category.

	precision	recall	f1-score	support
0	0.92	0.94	0.93	4696
1	0.94	0.92	0.93	4284
2	0.94	0.86	0.90	525
accuracy			0.93	9505
macro avg	0.93	0.91	0.92	9505
weighted avg	0.93	0.93	0.93	9505

Figure 4.74. Performance Evaluation of SVM with RBF

Linear SVM

The Linear-SVM, an algorithm renowned for its PREC and efficiency in high-dimensional spaces, was harnessed to classify news articles into 'fake', 'true', or 'half true' categories. Through the prism of a five-fold cross-validation methodology, the performance of the Linear-SVM has been meticulously analyzed to gauge its effectiveness in this complex, multi-class classification scenario.

The journey began with fold one, where the Linear-SVM set a promising precedent with a test ACC of 91.79%. Moving to fold two, the model showed slight improvement, achieving an ACC of 92.08%, indicative of its stable predictive prowess. In contrast, fold three experienced a marginal decrease in ACC to 91.63%, potentially highlighting the challenges posed by that particular subset of data or the intrinsic limitations of using a linear kernel when faced with complex, non-linear decision boundaries.

A significant spike in ACC was observed in fold four, where the model's performance soared to 94.68%. This peak represents the algorithm's potential in certain favourable conditions, possibly where the linear decision boundary aligns well with the data distribution. However, the fifth fold saw a recalibration to an ACC of 92.23%, aligning closer to the initial folds and providing a more conservative estimate of the model's average performance. The overall test ACC, determined after cross-validation, settled at 92.23%.

The CM generated for the linear SVM model provides insights into its robust performance. It offers a clear view of its ACC in classifying instances into the 'fake,' 'true,' and 'half true' categories (Figure 4.75). When examining the diagonal line running from the top left to the bottom right, representing the TP rates, it becomes evident that the model has achieved a high degree of correct predictions. This is particularly prominent in the 'fake' and 'true' categories, where the TP values reach 4374 and 3923, respectively. However, a closer inspection reveals the presence of misclassifications, as indicated by the off-diagonal elements. Notably, the 'true' category stands out with 357 instances misclassified as 'fake,' a significantly higher value than other misclassifications. This highlights an area that may benefit from refinement to enhance the model's ability to effectively distinguish between these two closely related classes.

Furthermore, the 'half true' category demonstrates a commendable TP rate 469. However, due to its smaller sample size, it experiences a slightly lower misclassification rate. This suggests a need to investigate further and improve the model's sensitivity to this category, aiming for a more balanced classification across all categories.

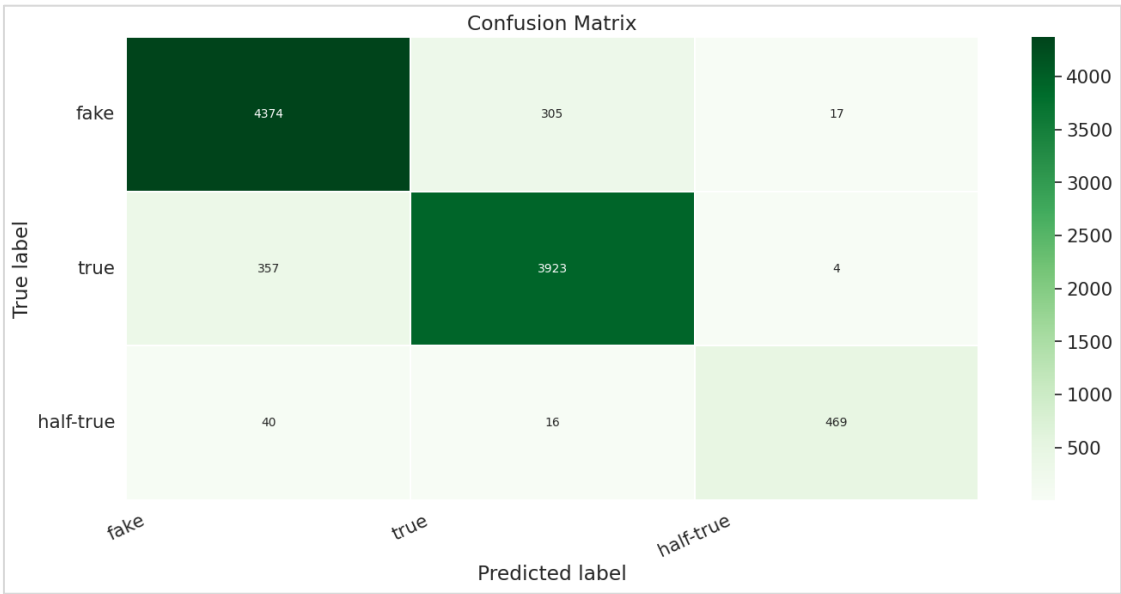


Figure 4.75. Evaluation Matrix for Linear SVM

The CR provides valuable insights into the performance of the linear SVM model, which exhibits a high level of efficiency with an overall ACC of 92.23% (Figure 4.76). When examining individual classes, it becomes evident that PREC and REC values consistently attain high levels for the 'fake' and 'true' categories. This underscores the model's proficiency in accurately classifying instances and its robustness in identifying relevant instances within these categories. Notably, the category representing 'half true' statements stands out with a PREC score of 0.96, the highest among the three categories. This indicates that when the model predicts a statement as 'half true,' it is highly likely to be correct, showcasing the model's strong discernment capability in recognizing the subtleties of this category. However, the REC for the 'half true' category is somewhat lower at 0.89, suggesting that while the model makes positive identifications for this class, it may miss some instances that should be categorized as 'half true.' Despite this, the F1-Ss remain consistently high across all categories, indicating that the model maintains a well-balanced and high-performance level in terms of both FP and FN for all classes. This performance nuance is particularly critical in the granularity of misinformation detection.

	precision	recall	f1-score	support
0	0.92	0.93	0.92	4696
1	0.92	0.92	0.92	4284
2	0.96	0.89	0.92	525
accuracy			0.92	9505
macro avg	0.93	0.91	0.92	9505
weighted avg	0.92	0.92	0.92	9505

Figure 4.76. Performance Evaluation of Linear SVM

Discussion

The comprehensive analysis of multiclass classification on Dataset 1 reveals a distinct performance hierarchy among the ML classifiers across three classes—Fake, Real, and Half-True. According to Table 4.5, RF shows consistent capability with an overall ACC (ACC) of 93.28%, maintaining a balance across PREC, REC, and F1-S, particularly excelling in the Half-True class with an F1-S of 91%.

GB excels with an ACC of 98.97%, demonstrating its strength with near-perfect scores, especially in identifying Real news with 100% PREC and REC. Its PREC and REC in the Half-True class are also robust at 93% and 97%, respectively, indicating its adeptness at nuanced classification.

XGBoost stands out with a stellar ACC of 99.56%, perfecting the Fake and Real news classification and exhibiting high PREC and REC of 98% in the Half-True category, asserting its dominance in complex classification scenarios.

Linear SVM, while slightly less accurate with an ACC of 92.22%, presents a balanced approach with consistent PREC and REC scores across all classes, particularly achieving a high PREC of 96% in the Half-True category.

The Radial Basis Function (RBF) SVM model, with an ACC of 92.99%, shows a similar trend to Linear SVM but slightly underperforms in the Half-True class with a PREC and REC of 94% and 86%, respectively.

LR nearly matches the top performers with an ACC of 98.99%, showcasing uniform excellence in classifying Fake and Real news and a high PREC of 99% for Half-True news, indicating its robustness across varied classifications.

The DT model closely follows the top contenders with an ACC of 99.27%, displaying near-perfect metrics in Fake and Real news classifications and strong PREC and REC of 97% for Half-True news, making it a reliable choice for complex classification tasks.

MNB, however, significantly underperforms with an ACC of 89.10%, particularly struggling with the Half-True class, showing 0% PREC and REC, which signals a critical limitation in handling ambiguous categories.

This evaluation underscores the exceptional effectiveness of advanced ensemble methods like GB and XGBoost in performing nuanced classifications while also highlighting the competency of RF and LR models. It also reveals the limitations of MNB, particularly in discerning the more ambiguous Half-True news category.

Table 4.5. ML classifiers result on dataset1

Classifier	ACC	Class	PREC	REC	F-Score
RF	93.28%	Fake	93%	94%	93%
		Real	94%	93%	93%
		Half-True	95%	88%	91%
GB	98.97%	Fake	99%	98%	99%
		Real	99%	100%	100%
		Half-True	93%	97%	95%
XGBoost	99.56%	Fake	100%	99%	100%
		Real	100%	100%	100%
		Half-True	98%	98%	98%
Linear SVM	92.22%	Fake	92%	93%	92%
		Real	92%	92%	92%
		Half-True	96%	89%	92%
RBF SVM	92.99%	Fake	92%	94%	93%
		Real	94%	92%	93%
		Half-True	94%	86%	90%
LR	98.99%	Fake	99%	99%	99%
		Real	99%	99%	99%
		Half-True	99%	95%	97%
DT	99.27%	Fake	99%	99%	99%
		Real	100%	100%	100%
		Half-True	97%	96%	96%
MNB	89.10%	Fake	89%	95%	92%
		Real	93%	93%	91%
		Half-True	0%	0%	0%

4.5.2. Deep learning classifiers results

LSTM

Throughout all five folds of the experiment, the LSTM model demonstrates a remarkable learning progression, as illustrated by the loss and ACC metrics trends during both the training and validation phases (Figure 4.77). Across multiple epochs, there is a consistent pattern of decreasing loss and increasing ACC, indicating successful model optimization with each successive epoch. In the initial epochs of each fold, there is a noticeable rapid decline in loss values and a significant increase in ACC metrics. For instance, in the first fold, the training loss drops from 0.5313 in the first epoch to 0.1201 in the final epoch, while ACC surges from 74.45% to 96.18%. Validation accuracies also remain commendably high, consistently staying above 93%. As the epochs progress in subsequent folds, the model reaches a peak of proficiency, notably showcased in the final fold. Here, the training loss decreases to an impressive 0.0031, and the ACC metric approaches near-perfection at 99.96%. Validation results mirror this excellence, with ACC reaching a peak of 99.84% in the fourth epoch. Test results across all folds further emphasize the robustness and high generalization capability of the LSTM model, with ACC consistently in the high 90s range. The fifth fold, in particular, achieves an impressive test ACC of 99.91%.

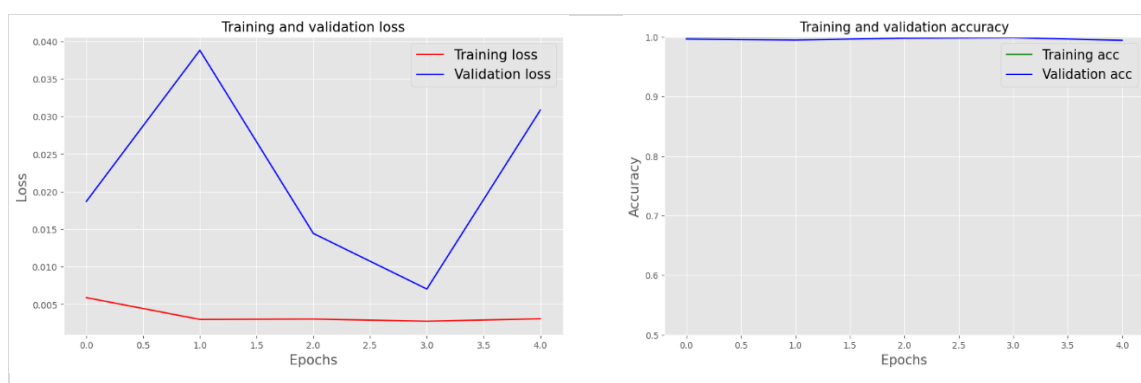


Figure 4.77. Accuracy and loss metrics for training and validation of LSTM

Upon examining the CM presented in Figure 4.78 for the LSTM model, it becomes evident that the model excels in accurately classifying the various labels: "fake," "true," and "half true." In the case of the "fake" label, out of 4705 instances, an impressive 4696 were correctly classified, with only a minimal number of misclassifications. Specifically, 8 instances were incorrectly labelled as "true," and a solitary instance was erroneously labelled as "half true."

The "true" category achieved a flawless classification rate, with all 4276 instances being precisely categorized without any FN or FP. Shifting the focus to the "half true" label, it also exhibits a strong predictive performance, correctly identifying all 524 instances, resulting in a perfect REC score for this class. While the results for the "true" and "half true" categories are unmatched, it is crucial to scrutinize the minor discrepancies observed in the "fake" label predictions to fine-tune the model further. Nonetheless, these results underscore the LSTM model's potent capability to distinguish between classes with high reliability and PREC, establishing it as a highly robust model for this classification task.

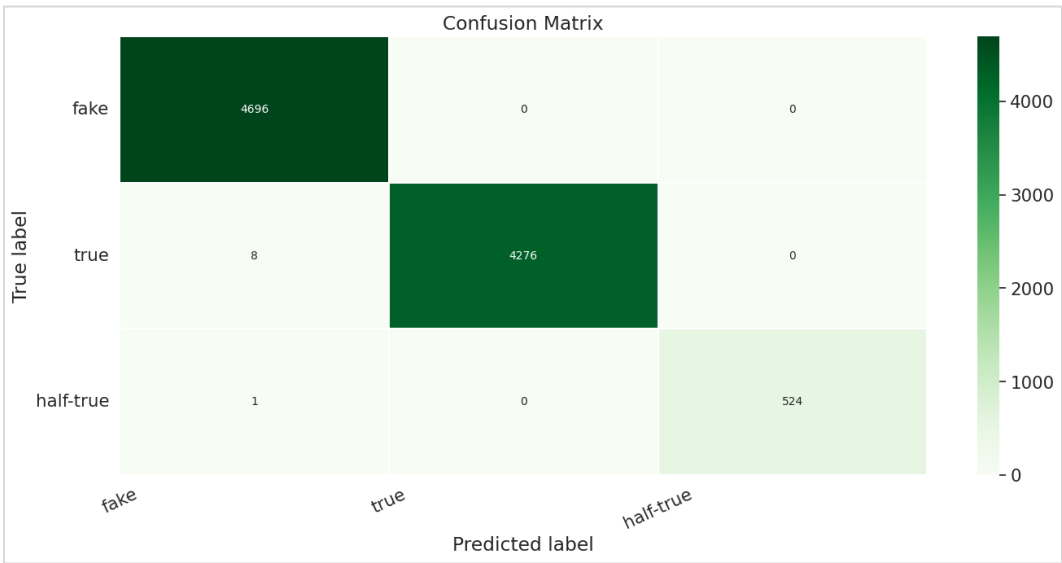


Figure 4.78. Evaluation Matrix for LSTM

The CR for the LSTM model showcases an almost flawless performance across all the assessed metrics, PREC, REC, and F1-S, pertaining to each of the categories labelled 0, 1, and 2 (Figure 4.79). In every one of these categories, the model attained a perfect score of 1.00 for all the aforementioned metrics, underscoring its ability to accurately identify TP without erroneously assigning other instances to these categories. This remarkable performance is further substantiated by the ACC metric, which registers at 0.9991 on the testing set. This high ACC value illustrates that the LSTM model has achieved nearly impeccable ACC in distinguishing between the various categories in this specific task. Additionally, both the macro and weighted averages of the PREC, REC, and F1-S also hold a perfect score of 1.00. This reiterates the model's supreme performance in managing the classifications' individual and collective ACC, emphasizing its exceptional proficiency in this classification task.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4705
1	1.00	1.00	1.00	4276
2	1.00	1.00	1.00	524
accuracy			1.00	9505
macro avg	1.00	1.00	1.00	9505
weighted avg	1.00	1.00	1.00	9505

Figure 4.79. Performance Evaluation of LSTM

CNN-LSTM

Analyzing the outcomes of a k-fold cross-validation training involving 5 folds for a CNN-LSTM hybrid model, we can discern a striking upward trajectory in the ACC metric accompanied by a simultaneous decrease in the loss metric throughout training epochs (Figure 4.80). In the initial epoch of the first fold, the training ACC kicks off impressively at 75.61%, and this performance consistently improves throughout subsequent epochs, culminating in an exceptional training ACC of 97.27% at the end of the fifth epoch. A similar pattern emerges in the validation ACC, commencing strongly at 92.35% and steadily ascending to an outstanding 99.53%. Simultaneously, the training and validation loss metrics significantly decline, indicating that the model becomes highly optimized by the conclusion of the first fold's iterations. The model builds upon the optimization achieved in prior folds in each subsequent fold, resulting in even higher accuracies and lower loss metrics in both the training and validation sets. In fold 2, the initial training ACC spikes to 97.48% and maintains an upward trajectory, reaching 99.46% by the final epoch. The validation subset shows a similar pattern, with accuracies consistently above 99%. Test accuracies also experienced a substantial increase, climbing from 96.83% in fold 1 to 99.51% in fold 2. As we progress through folds 3, 4, and 5, the model continues to excel, with initial epoch accuracies consistently starting around 99% and maintaining this high level of performance. By the end of the final epoch of fold 5, the training ACC reaches an impressive 99.87%, as evidence of the model's effective learning. Validation and test accuracies follow suit, demonstrating a steadfast ascent and peaking at an almost impeccable ACC rate of 99.91% in fold 5. Furthermore, the loss metrics mirror this optimisation narrative, with training, validation, and test loss metrics all experiencing a noticeable decline across epochs and folds. This suggests that the model has finely honed its predictive capabilities to a highly refined state.

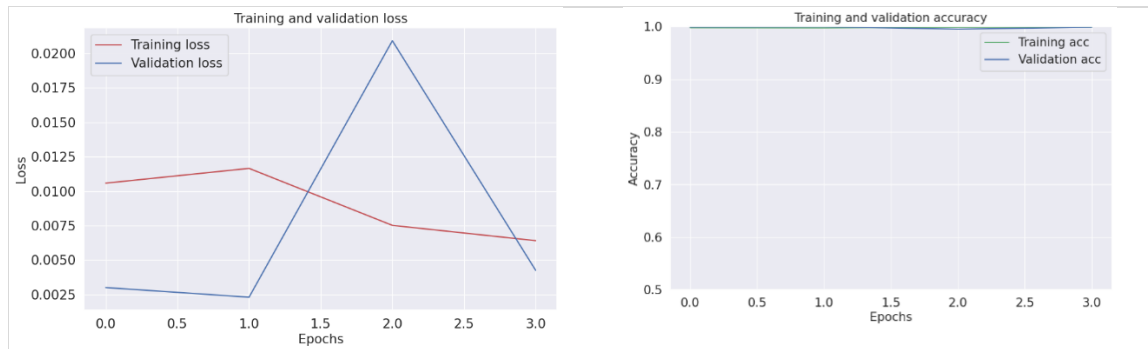


Figure 4.80. Accuracy and loss metrics for training and validation of CNN-LSTM

Analyzing the CM that depicts the CNN-LSTM model's predictions provides a comprehensive assessment of the model's performance across the distinct classes: "fake," "true," and "half true" (Figure 4.81). The diagonal running from the top left to the bottom right, which signifies correctly classified instances, showcases a praiseworthy performance. It comprises values of 4691, 4284, and 521 for the "fake," "true," and "half true" classes, respectively. These values indicate that the model adeptly identified the majority of instances within each class, underscoring a high level of ACC in its predictions. Furthermore, the "true" class stands out with a perfect REC score, featuring 4284 correct predictions and zero misclassifications. This perfect REC score signifies the model's impeccable ability to identify instances in this category accurately. However, there were a few instances where the model made minor errors, albeit in a negligible proportion. Specifically, it misclassified four instances as "half true" instead of "fake" and five instances as "fake" rather than "half true."

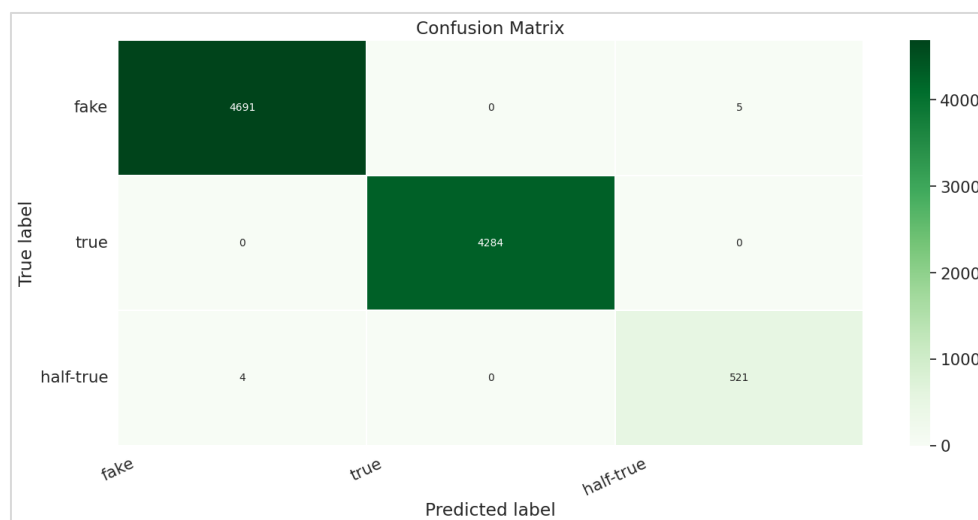


Figure 4.81. Evaluation Matrix for CNN-LSTM

Upon reviewing the CR that assesses the CNN-LSTM model's performance, it becomes evident that the model demonstrates an exceptionally high level of ACC, achieving a score of 99.9% on the testing set (Figure 4.82). This exceptional ACC is further substantiated by the consistently high PREC, REC, and F1-S scores for each of the individual classes. Specifically, classes 0 and 1 exhibit perfect scores across all these metrics, underscoring the model's exceptional sensitivity and PREC in identifying instances within these categories. When closely examining class 2, a marginal decrease in performance is observed compared to the other categories, with PREC, REC, and F1-S all standing at 99%. While this may suggest a slightly reduced ability to identify instances in this category compared to others accurately, it's crucial to emphasize that a 99% score still signifies a remarkably high level of performance. This highlights the model's capability to perform well even in the least favourable scenario considered in this analysis. Furthermore, the model's outstanding performance is consistently reinforced when considering the macro average and weighted average metrics across all categories and evaluation criteria. These metrics either reach or closely approach a score of 1.00, indicating an almost flawless ability to classify instances across all categories correctly.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4695
1	1.00	1.00	1.00	4284
2	0.99	0.99	0.99	526
accuracy			1.00	9505
macro avg	1.00	1.00	1.00	9505
weighted avg	1.00	1.00	1.00	9505

Figure 4.82. Performance Evaluation of CNN-LSTM

CNN

When examining the results of the CNN model's training and validation process across five folds, it becomes evident that there is a fluctuation in both ACC and loss throughout different epochs, signifying varying performance during the iterative process (Figure 4.83). In the initial fold, the model starts with a relatively high loss, gradually decreasing throughout the epochs, eventually converging at a loss of 0.731. Correspondingly, the ACC reaches around 57.1%. This trend of decreasing loss continues to some extent in fold 2, although there is a spike in the validation loss in the final epoch, which could indicate potential overfitting.

Moving on to the third fold, the ACC hovers around 59.9%, demonstrating a stable, albeit not significantly improved, performance across epochs. The loss metric exhibits a slight tendency to decrease but with fluctuations. The fourth fold presents a notable deviation from the previous folds, characterized by a surge in the validation loss in the first, third, and fourth epochs.

Additionally, there is a substantial dip in validation ACC in the third epoch, dropping to 22.95%. These observations raise concerns regarding the model's stability and ability to generalize well for unseen data, suggesting a phase where the model may have diverged. The model showcases a more stabilized performance in the fifth fold, with the loss progressively diminishing. This indicates a refinement in the model's predictive abilities, reflected in the test ACC. While it may not be the highest observed, it suggests a consistency level in the model's predictive power.



Figure 4.83. Accuracy and loss metrics for training and validation of LSTM

The provided CM offers insights into CNN's performance when classifying instances into three categories: "fake," "true," and "half true" (Figure 4.84). A closer examination of the results reveals that the model excels in predicting the "true" category, correctly identifying 3064 instances. However, it also produces a notable number of FP, totalling 2408. It's worth highlighting that the "fake" category shows a wider range of results, with 2124 TP. However, it exhibits a significant number of instances being misclassified as "true" (1216 instances) and, to a lesser extent, as "half true" (68 instances). In the case of the "half true" category, which has the fewest instances, the model demonstrates reasonable PREC by correctly predicting 453 instances. However, it struggles with REC, as a considerable number of "half true" instances are misclassified as both "fake" and "true." These observations suggest that

while the model performs well in identifying "true" statements, it faces challenges in accurately classifying "fake" and "half true" statements, leading to a notable presence of FP and FN in these categories.

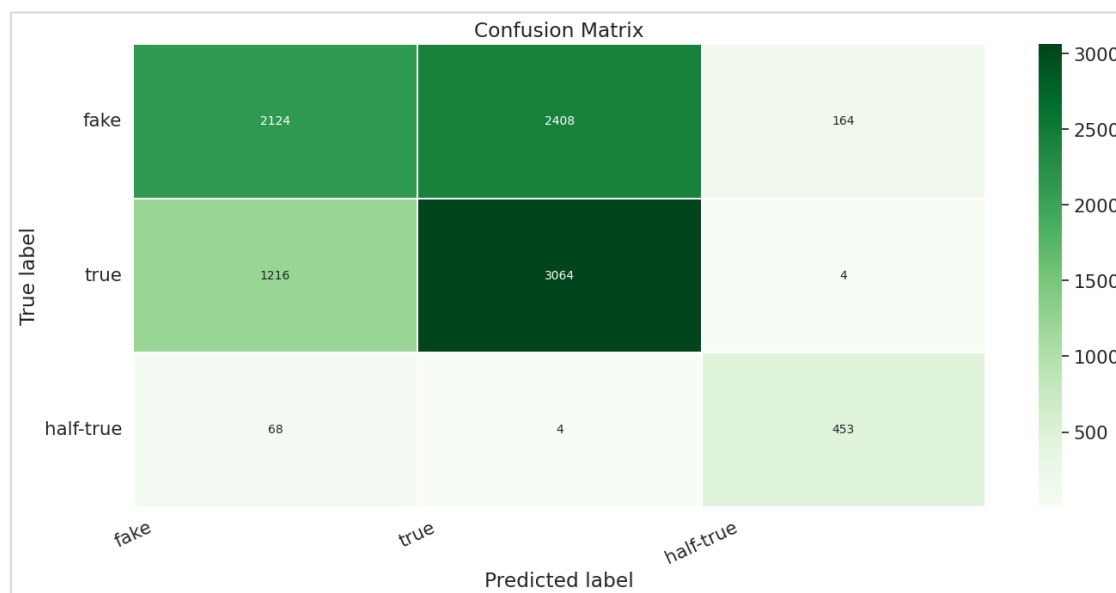


Figure 4.84. Evaluation Matrix for CNN

Upon analyzing the results derived from the CM and CR, it becomes evident that the CNN model showcased varying performance across different metrics and classes (Figure 4.85). For class 0, which represents the "fake news" category, the PREC score is 0.45. This indicates that among all the instances predicted as class 0 by the model, only 45% were correctly classified. However, it exhibits a higher REC score of 0.62, signifying that the model correctly identified 62% of all actual class 0 instances. The F1-S, which harmonizes PREC and REC, stands at 0.52 for this class.

In contrast, class 1 displayed a higher PREC of 0.72 but a lower REC of 0.56, resulting in an F1-S of 0.63. This suggests that while the model was relatively precise in identifying this class, it failed to recognize a substantial portion of actual class 1 instances. Class 2 demonstrated the highest PREC and F1-S among the three classes, with values of 0.86 and 0.79, respectively, along with a REC of 0.73. This signifies that the model was exact and moderately sensitive regarding class 2 instances. The macro average and weighted average metrics offer a generalized view of the model's performance, indicating a moderate level of efficiency with scores ranging from 0.64 to 0.68 for the macro average and 0.59 to 0.63 for the weighted average. The overall ACC of the model on the test set is 59.35%, implying that

it correctly predicted the class for nearly 59.35% of the total instances. While the model demonstrates considerable proficiency, especially in classifying class 2 instances, there is room for improvement, particularly in enhancing the PREC for class 0 and improving the REC for class 1.

	precision	recall	f1-score	support
0	0.45	0.62	0.52	3408
1	0.72	0.56	0.63	5476
2	0.86	0.73	0.79	621
accuracy			0.59	9505
macro avg	0.68	0.64	0.65	9505
weighted avg	0.63	0.59	0.60	9505

Figure 4.85. Performance Evaluation of CNN

LSTM Autoencoder

While training and validating the LSTM-AutoEncoder model across multiple folds, a nuanced pattern emerges regarding ACC and loss (Figure 4.86). The model demonstrates a relatively stable performance throughout five epochs per fold, with both training and validation ACC hovering around the 45% mark. This suggests that the model struggles to improve its predictive power and raises the possibility of an underfitting issue. While there are minor fluctuations in validation loss and ACC across the epochs, the overall trend remains consistent, indicating that the model's performance doesn't change substantially with each additional epoch. In the initial epoch of the first fold, the model achieves a training ACC of 45.54% with a loss of 0.7365. In the validation phase, it reaches an ACC of 45.15% with a loss of 0.6864. Subsequent epochs maintain stable training and validation accuracies around 45%, with validation losses showing slight variations but generally staying below 0.7. As the LSTM-AutoEncoder model progresses through additional folds, it maintains a consistent pattern of stable yet suboptimal performance. Test accuracies after the fifth epoch in each fold consistently remain in the 45% range, indicating a lack of improvement with additional epochs.

Similarly, test loss values do not exhibit a consistent declining trend over the epochs, fluctuating between 0.6849 and 0.6868, further suggesting a lack of model optimization.

This recurrent theme persists across all folds, with no significant enhancements in ACC or reductions in loss as the epochs progress. Final test accuracies consistently fall between 45.07% and 45.09%, underscoring a noticeable lack of development over time. Test losses mirror this pattern, oscillating within a narrow range from 0.6849 to 0.6867.

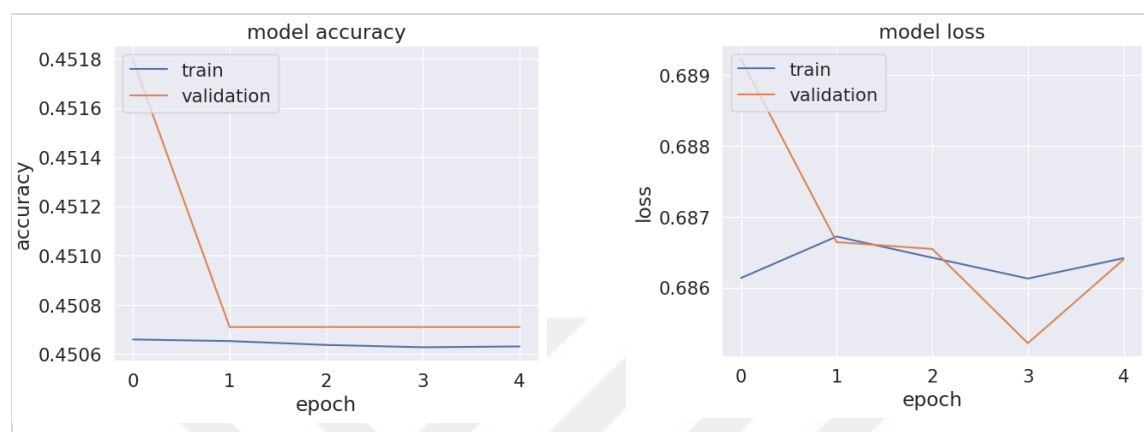


Figure 4.86. Accuracy and loss metrics for training and validation of LSTM-AutoEncoder

The CM for the LSTM-AutoEncoder model presents the results of its performance on a three-class classification task—Fake, True, and Half-True. The model demonstrates high ACC in identifying fake news, with 4,695 correct predictions and no misclassifications (Figure 4.87). Similarly, it accurately classifies all 4284 instances of True News without error. However, the model struggles with the Half-True category, correctly identifying only 525 instances. It does not misclassify any of the True or Fake news as Half-True. Still, it also fails to recognize any Half-True instances beyond the 525 correctly identified, as indicated by zeroes in the columns for True and Fake under the Half-True row. This suggests that while the model is highly effective at discerning clear-cut Fake and True instances, its ability to detect nuances and classify news as Half-True is severely limited, potentially due to a lack of distinctive features within the data or insufficient model complexity to capture the subtleties of this intermediate class.

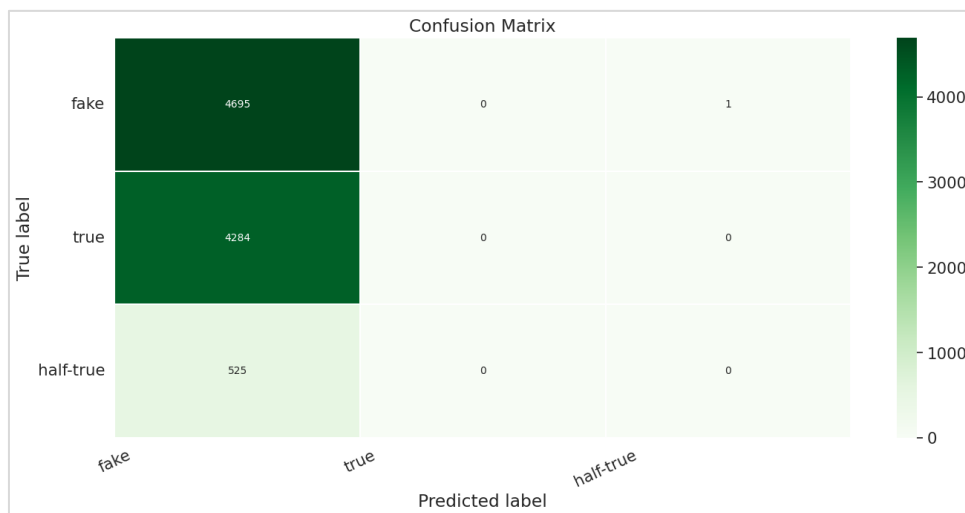


Figure 4.87. Evaluation Matrix for LSTM-AutoEncoder

The results of the LSTM-AutoEncoder model, presented in terms of PREC, REC, and F1-S, highlight some significant limitations in its performance (Figure 4.88). For the fake (class 0) articles, the model achieved a perfect PREC of 100%, indicating that it was correct when it predicted an article to be fake. However, the REC for this class is considerably low at 49%, suggesting that the model missed a substantial number of fake articles in the dataset. The F1-S for class 0 also stands at 66%, reflecting the trade-off between PREC and REC.

For the Real (class 1) articles, the PREC, REC, and F1-S are all at 0%. This indicates that the model failed to correctly classify any articles as accurate, resulting in complete misclassification for this class.

The Half-True (class 2) articles also experienced similar performance issues, with PREC, REC, and F1-S at 0%. In terms of overall ACC, the model achieved an ACC of 49%, which reflects the combined performance across all classes.

The LSTM-AutoEncoder model demonstrates a significant deficiency in classifying real and half-true articles, with very low REC and F1-Ss for these classes. While it excels in PREC for the fake class, it struggles to achieve a balanced performance across all categories, resulting in an overall ACC of 49%.

	precision	recall	f1-score	support
0	1.00	0.49	0.66	9504
1	0.00	0.00	0.00	0
2	0.00	0.00	0.00	1
accuracy			0.49	9505
macro avg	0.33	0.16	0.22	9505
weighted avg	1.00	0.49	0.66	9505

Figure 4.88. CR of LSTM-AutoEncoder

CNN-GRU

In the conducted research, the CNN-GRU model underwent rigorous training and validation using a five-fold strategy to ensure robustness and reliability in predictive ACC (Figure 4.89). During the first fold, the model exhibited a remarkable improvement in ACC, starting at 84.38% in the initial epoch and reaching an impressive 99.94% in the final epoch. Concurrently, it reduced the loss from 0.3075 to a mere 0.0028. The validation segments consistently showed high ACC, exceeding 99%, with the final test resulting in an ACC of 99.67% and a loss of 0.0179. The model continued to enhance its performance in the second fold, maintaining ACC above 99.86% throughout the training epochs. The validation phase mirrored this high ACC, peaking at 99.93% in the fourth epoch. The test results for this fold demonstrated an astonishing ACC of 99.89% with a minimal loss of 0.0039. Moving to the third fold, the model displayed elite performance, with validation ACC consistently between 99.82% and 99.97%, indicating its exceptional ability to generalize on unseen data. The final testing phase reported a loss of 0.0038 and an impressive ACC of 99.93%. In the fourth fold, the model sustained its superior performance, achieving near-flawless ACC of up to 100% in the validation set of the fourth epoch, demonstrating its nearly perfect learning. The final testing phase concluded with a marginal loss of 0.00098 and a breathtaking ACC of 99.98%. Entering the final fold, the CNN-GRU model maintained its momentum, showing stability with ACC consistently exceeding 99.81% in the validation phases. The ultimate testing epoch recorded a minimal loss of 0.0020, confirming a high ACC of 99.97%.

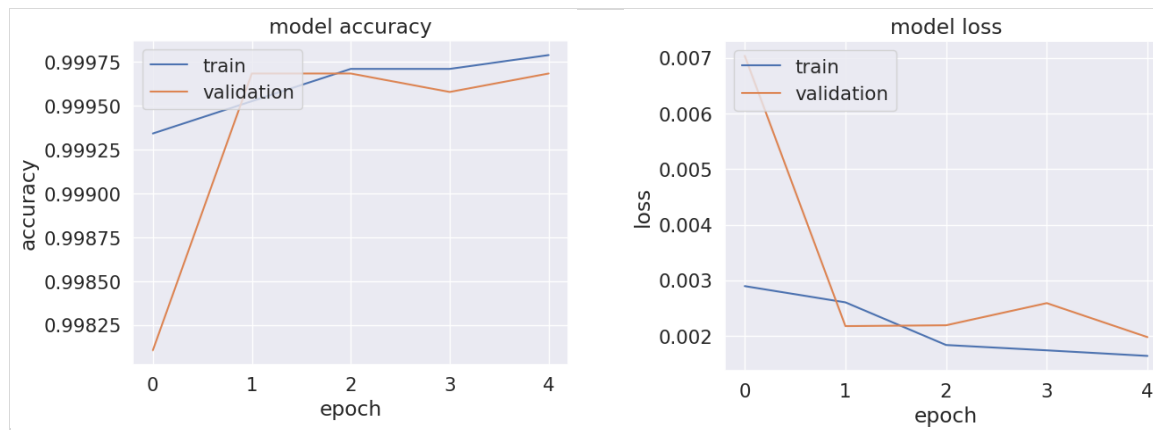


Figure 4.89. Accuracy and loss metrics for training and validation of LSTM

In assessing the CNN-GRU model's performance using the CM, we can observe a high degree of ACC in classifying instances into the 'fake,' 'true,' and 'half-true' categories (Figure 4.90). Specifically, the 'fake' category had 4694 correct predictions with only 2 misclassifications as 'half-true,' and no instances were incorrectly labelled as 'true.' The 'true' class yielded 4283 accurate predictions with a single misclassification into the 'half-true' category. In the case of the 'half-true' classification, all 525 instances were correctly identified, with no misclassifications into other categories. However, it's noteworthy that both the 'fake' and 'true' categories had minimal instances misclassified as 'half-true.' This high level of ACC in classification, as indicated by the diagonal elements of the CM, underscores the CNN-GRU model's PREC and REC, signifying its successful performance in distinguishing between 'fake,' 'true,' and 'half-true' designations.

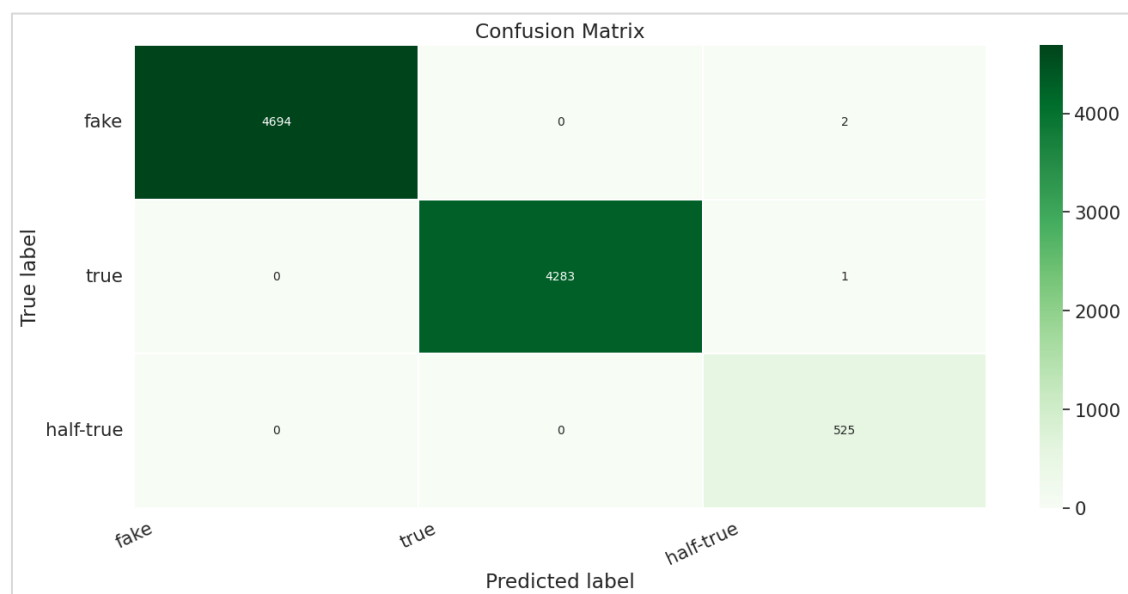


Figure 4.90. Evaluation Matrix for CNN-GRU

Examining the performance metrics of the CNN-GRU model as presented in the CM, we can observe an exceptional level of prediction ACC across all classes (Figure 4.91). The PREC, REC, and F1-S metrics, crucial indicators of a model's performance, demonstrate near-perfect scores across the categories labelled 0, 1, and 2. Both classes 0 and 1 achieved flawless scores in PREC, REC, and F1-S, signifying that the model could identify these classes without errors. Meanwhile, class 2 exhibited slightly lower, though still highly high, REC and F1-S values, respectively, at 0.99. The minor reduction in REC and F1-S for class 2 suggests a slight potential for improving the model's sensitivity to identify all relevant instances for this class correctly.

Additionally, the ACC metric rounds off to 1.00 when taken to two decimal places, precisely holding a value of 0.9997. This underscores the model's robust performance on the testing set, which consists of 9505 instances. In conclusion, the results demonstrate that the CNN-GRU model has achieved a highly satisfactory level of performance, effectively balancing PREC and REC, as evidenced by the harmonized F1-S.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4694
1	1.00	1.00	1.00	4283
2	1.00	0.99	1.00	528
accuracy			1.00	9505
macro avg	1.00	1.00	1.00	9505
weighted avg	1.00	1.00	1.00	9505

Figure 4.91. Performance Evaluation of CNN-GRU

Discussion

Our exploration of DL techniques for the multiclass classification of FND (FND), which involves three classes—fake (0), true (1), and half-true (2), yielded remarkable results. Among the evaluated algorithms, the LSTM model stood out with unparalleled performance, achieving an ACC (ACC) of 99.90%. Notably, it recorded perfect PREC, REC, and F1-S scores across all classes in Table 4.6. This exceptional level of performance underscores the LSTM model's capacity to differentiate between the nuanced categories of truthfulness effectively. The CNN-LSTM model closely matched the LSTM's performance, achieving an

ACC of 99.90%. The model demonstrated perfection in PREC, REC, and F1-S for classes 0 and 1 while nearly reaching perfection with 99% scores across the metrics for class 2. This suggests that the fusion of CNN and LSTM components offers a robust solution for this complex classification task.

In contrast, the CNN-GRU model outperformed all others with an astonishing ACC of 99.96%. It displayed nearly perfect metrics across the board, achieving a perfect 1.00 in all metrics for classes 0 and 1, with only a slight deviation in REC for class 2, which scored 0.99. The exceptional performance of this model highlights the effectiveness of combining CNN with GRUs (Gated Recurrent Units) in capturing the complexities involved in FND. On the other hand, the standalone CNN model exhibited significant underperformance, achieving an ACC of only 59.34%. This model displayed noticeable inconsistencies in metrics across classes, with PREC ranging from a low of 45% to a high of 86% and REC varying from 56% to 73%. These disparities suggest that while CNNs may excel in certain tasks, they appear inadequate for this specific multiclass classification challenge. Finally, the LSTM AutoEncoder delivered a concerning result, with an ACC of just 49%. It scored 100% in PREC but only 49% in REC for the 'Fake' class and failed for the 'True' and 'Half-True' classes, recording zeros across all metrics. This outcome raises questions about the suitability of the AutoEncoder architecture for this task and may indicate issues with overfitting or training data representation.

Table 4.6. DL classifiers result on dataset1

Classifier	ACC	Class	PREC	REC	F-Score
CNN	59.34%	Fake	45%	62%	52%
		Real	72%	56%	63%
		Half-True	86%	73%	79%
CNN-LSTM	99.90%	Fake	100%	100%	100%
		Real	100%	100%	100%
		Half-True	99%	99%	99%
CNN-GRU	99.96%	Fake	100%	100%	100%
		Real	100%	100%	100%
		Half-True	100%	99%	100%
LSTM	99.90%	Fake	100%	100%	100%
		Real	100%	100%	100%
		Half-True	100%	100%	100%
LSTM AutoEncoder	49%	Fake	100%	49%	66%
		Real	0%	0%	0%
		Half-True	0%	0%	0%

4.6. Results Using Dataset2

4.6.1. Machine learning classifier results

Random forest

Our implementation of the RF algorithm has demonstrated considerable efficacy in the quest to discern the veracity of news items, as evidenced by a series of cross-validated tests. The algorithm classified news into four categories: 'fake', 'true', 'half true', and 'barely true'. Across five distinct folds intended to validate the model's robustness and consistency, the results were commendable and reflected a high degree of ACC. The first fold revealed an ACC of 86.96%, a promising start that set a positive precedent for the subsequent evaluations. Fold two witnessed a slight uptick in performance, achieving an ACC of 88.17%. The third fold further reinforced the model's proficiency, cresting at an ACC of 88.87%, the zenith of the test folds. A modest decline was observed in fold four, with ACC marginally receding to 87.58%. However, fold five mirrored the performance of fold two, with an ACC of 88.25%. These folds culminated in an aggregated test ACC score of 88.25%, as confirmed by the `ACC_score` function post the cross-validation process. This score substantiates the RF model's capacity to classify news articles with a high degree of ACC, showcasing its potential for navigating the complex landscape of news veracity. The consistency across the folds highlights the model's stability and reliability, marking it as a robust classifier in FND.

Upon analyzing the CM for the RF model, we can observe distinct patterns in its classification performance across the categories of "fake," "true," "half true," and "barely true" (Figure 4.92). A closer look at the diagonal, representing correct predictions, reveals that the model has achieved a high level of ACC in differentiating between the "fake" and "true" categories, with 4428 and 3896 correct predictions, respectively. However, when we shift our focus to the "half true" and "barely true" classifications, we notice a decrease in ACC, accompanied by a higher number of misclassifications. This trend is particularly pronounced in the case of "half true," where the model made 309 correct predictions but also exhibited notable numbers of instances erroneously classified into other categories. A pattern for the "barely true" category is observed, with 126 correct predictions. These results suggest that while the model demonstrates strong efficacy in distinguishing between the more

straightforward "fake" and "true" classifications, it tends to become confused when tasked with discerning the more nuanced "half true" and "barely true" categories. This confusion is reflected in the relatively scattered distribution of predictions for these categories across other labels.

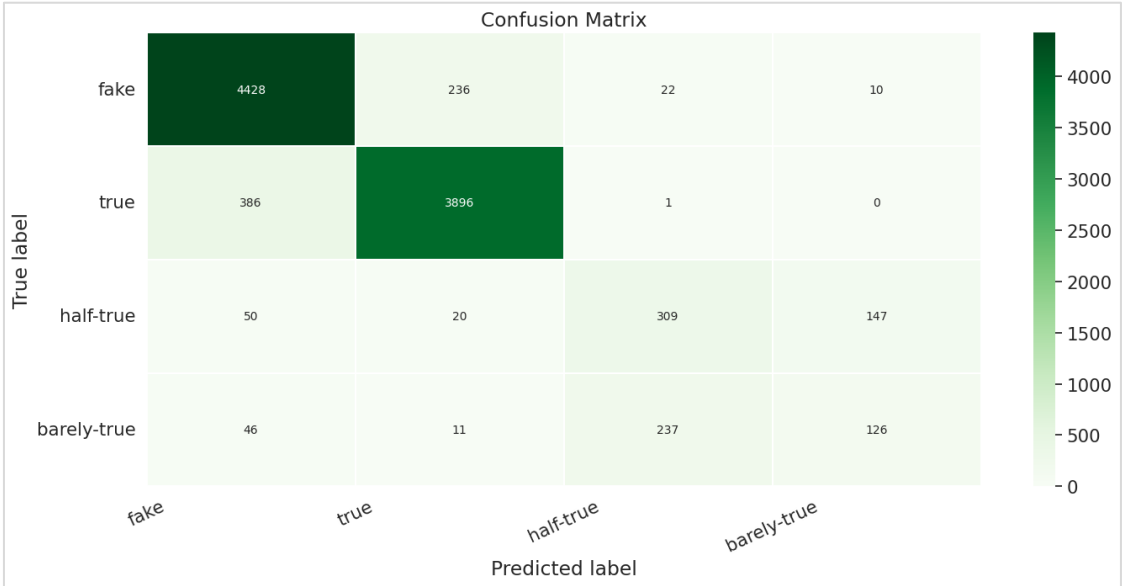


Figure 4.92. Evaluation Matrix for RF

Upon analyzing the results presented in the CM for the RF model, we can discern several noteworthy trends in its predictive performance across four distinct classifications (0, 1, 2, 3) based on various metrics, including PREC, REC, and F1-S (Figure 4.93). In categories 0 and 1, the model demonstrates a high level of PREC, scoring 0.90 and 0.94, respectively. This indicates a consistent ability to identify instances belonging to these categories accurately. The strong F1-Ss of 0.92 for both categories further emphasize a balanced performance, considering both PREC and REC. The REC rates of 0.94 and 0.91 for categories 0 and 1, respectively, signify the model's effectiveness in correctly identifying most actual positives.

Conversely, when we shift our focus to categories 2 and 3, we observe a noticeable decline in the model's performance. PREC, REC, and F1-Ss in these categories are lower, with PREC scores of 0.54 and 0.45 and REC scores of 0.59 and 0.30 for categories 2 and 3, respectively. The relatively lower F1-Ss of 0.56 and 0.36 in these categories indicate a challenge in maintaining a balanced performance. The overall ACC of the model is 88.25%, with a macro average of around 0.69 for both PREC and REC. This suggests a moderate

level of effectiveness in generalizing across all categories. However, the notable disparities in performance metrics between the categories highlight areas where improvement may be beneficial, particularly in enhancing the classification ACC for categories 2 and 3.

	precision	recall	f1-score	support
0	0.90	0.94	0.92	4696
1	0.94	0.91	0.92	4283
2	0.54	0.59	0.56	526
3	0.45	0.30	0.36	420
accuracy			0.88	9925
macro avg	0.71	0.69	0.69	9925
weighted avg	0.88	0.88	0.88	9925

Figure 4.93. Performance Evaluation of RF

Gradient boosting

The application of GB for classifying news into 'fake', 'true', 'half true', and 'barely true' has yielded exceptionally high ACC rates across a five-fold cross-validation framework.

Commencing with fold one, the GB model demonstrated a high level of discernment with an ACC of 94.76%. This initial success was slightly surpassed in fold two, where the model achieved an ACC of 95.05%, showcasing its capability to navigate the intricacies of fake news classification effectively. Fold three maintained this high standard of ACC, registering a fractionally lower yet substantial 94.96%. A negligible variation was observed in fold four, with an ACC of 94.71%, which still represents a high level of classification performance. The fifth and final fold confirmed the algorithm's consistently high performance, concluding with an ACC of 94.89%.

Moreover, the overall test ACC, evaluated post-cross-validation, resonated with the results of the final fold, standing firmly at 94.89%. This figure not only corroborates the individual fold accuracies but also highlights the robustness and reliability of the GB classifier in distinguishing varying degrees of truthfulness in news content. The aggregated results, marked by their high PREC, reinforce the suitability of GB in handling FND's complex and nuanced task. The model's ability to sustain such high ACC across all folds suggests an

excellent generalization capability, marking it as a potent analytical tool in the ongoing effort to ensure the integrity of information in the digital age.

Upon analyzing the CM for the RF model, it becomes evident that the model excels in distinguishing between the "fake" and "true" categories. This is reflected in the high TP counts of 4622 and 4275 for these respective categories (Figure 4.94). Furthermore, these categories exhibit minimal FN, resulting in high REC values. The matrix also reveals a remarkable level of PREC in classifying "true" instances. With a TP count of 4275 and a negligible false positive count, the model achieves an almost perfect PREC score in this category. However, the model's performance becomes more mixed when we shift our focus to the "half true" and "barely true" categories. The "half true" category shows a substantial TP count of 399, but it also includes a notable number of instances incorrectly classified as "barely true." Similarly, in the case of the "barely true" category, although it maintains a healthy TP count of 122, a significant number of instances are mistakenly classified as "half true." This suggests some confusion exists between these categories in the model's predictions.

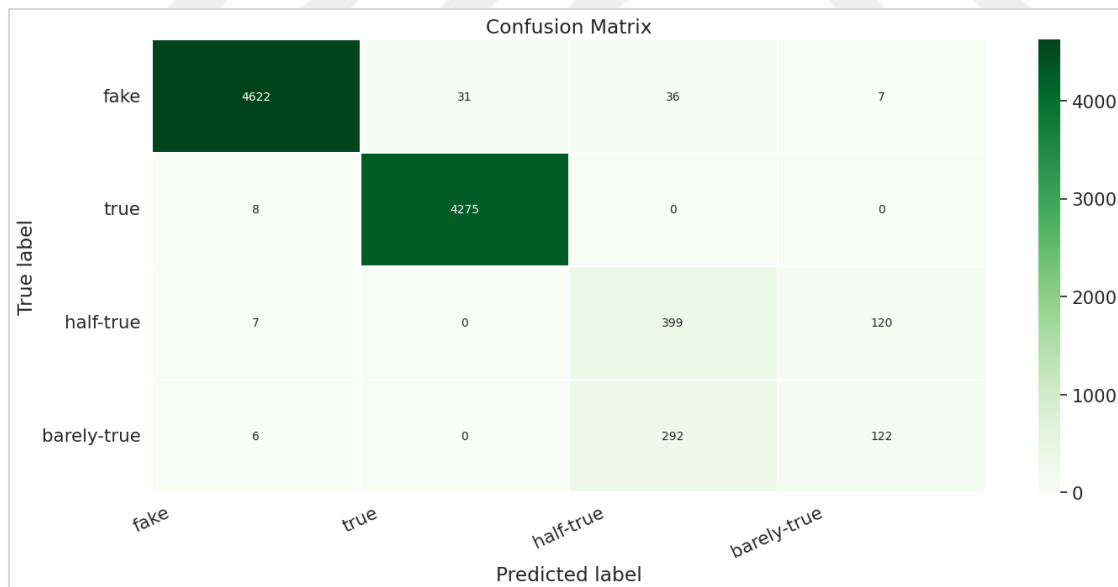


Figure 4.94. Evaluation Matrix for GB

Upon analyzing the results derived from the CM for the GB model, we observe its commendable performance in classifying categories "0" and "1," as evidenced by the PREC and REC values (Figure 4.95). In the case of category "0," the model achieves a remarkably high PREC score of 1.00, indicating that it makes very few false positive predictions. While

slightly lower at 0.98, the REC is still quite high, signifying the model's proficiency in correctly identifying most instances in this category with minimal FN. Category "1" exhibits a similar level of excellence, with a perfect REC score of 1.00 and a near-perfect PREC score of 0.99. This implies that the model excels in PREC and REC for this category, making very few misclassifications. However, the model faces more complexity when dealing with categories "2" and "3." For category "2," the PREC is at 0.55, indicating that while the model identifies many TP, it also misclassifies many instances from other categories belonging to category "2." On the other hand, the REC for category "2" is relatively higher at 0.76, suggesting that the model captures most of the actual category "2" instances. Category "3" presents a different challenge, with a PREC slightly lower than Category "2" at 0.49 and a significantly lower REC of 0.29. This indicates that the model misclassifies a substantial portion of actual category "3" instances. In conclusion, the GB model's primary strength lies in its ability to accurately classify categories "0" and "1," which significantly contributes to its overall ACC of 0.95. However, it faces challenges in classifying categories "2" and "3," with room for improvement in PREC and REC.

	precision	recall	f1-score	support
0	1.00	0.98	0.99	4696
1	0.99	1.00	1.00	4283
2	0.55	0.76	0.64	526
3	0.49	0.29	0.36	420
accuracy			0.95	9925
macro avg	0.76	0.76	0.75	9925
weighted avg	0.95	0.95	0.95	9925

Figure 4.95. Performance Evaluation of GB

Logistic Regression

The deployment of LR in our study to classify news statements into 'fake', 'true', 'half true', and 'barely true' has provided remarkably consistent and high ACC rates through a five-fold cross-validation scheme. Traditionally renowned for binary classification, this statistical method was adeptly extended to handle our multi-class classification challenge with notable success.

Our first fold set a strong precedent with an ACC of 94.68%. This was closely followed by the second fold, which showed a slight improvement, presenting an ACC of 94.88%. The third fold offered a comparable outcome with an ACC of 94.80%, suggesting a stable LR model performance across varying data subsets. The fourth fold registered the highest ACC, reaching 95.08% and hinting at the model's capacity to effectively generalize from the training data. The fifth fold saw a minor dip to an ACC of 94.91%, but this remained in line with the high standards set by the previous folds. The overall test ACC, computed after the cross-validation process, stood at 94.91%, which resonates with the high accuracies observed in the individual folds.

In the CM depicted in Figure 4.96, corresponding to the LR model shown in Figure 4.91, we can discern notable trends in terms of PREC and REC, particularly in the classification of "fake" and "true" labels. The majority of predictions fall within the TP quadrant for these categories, indicative of a high level of predictive ACC, especially for the more clearly defined ends of the truth spectrum. To elaborate, the "fake" category exhibits a robust TP rate, with 4645 out of 4696 instances correctly classified. The presence of only a small number of FP distributed among the other categories underscores the model's effectiveness in distinguishing "fake" instances. Similarly, the "true" category showcases a high TP rate, correctly identifying 4252 out of 4283 instances. This category also experiences a few misclassifications, further affirming the model's PREC in classifying "true" instances. However, the LR model encounters greater challenges when accurately predicting the "half true" and "barely true" categories. We observe a more substantial spread of FP and FN in these cases. For instance, the "half true" category demonstrates a TP rate of 368, but it also exhibits a significant number of instances misclassified as "barely true" (137). Likewise, the "barely true" category witnesses a notable portion of its actual instances being misclassified, particularly into the "half true" and "fake" categories.

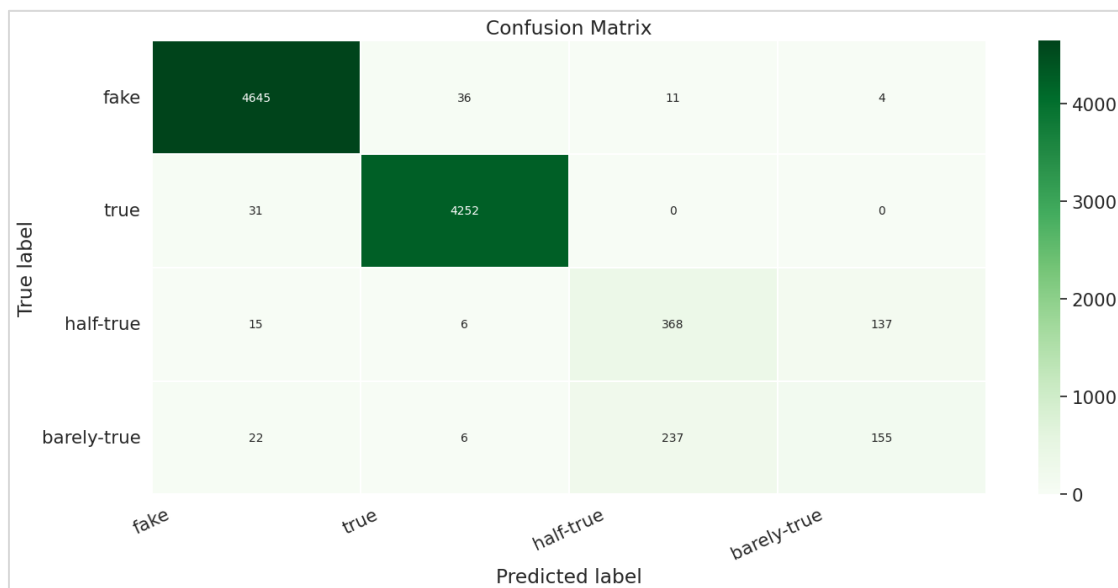


Figure 4.96. Evaluation Matrix for LR

The CM provided for the GB model reveals exceptional levels of PREC and REC in the "0" and "1" categories, nearly reaching perfection with scores of 0.99 for both PREC and REC metrics (Figure 4.97). This underscores the model's remarkable ability to accurately classify instances within these two categories, highlighting its substantial predictive strength and dependability for these classes. However, as we shift our focus to the "2" and "3" categories, the model's performance experiences a decline. In category "2", PREC and REC decrease to 0.60 and 0.70, respectively, indicating a moderate ability to correctly identify TP within this group. In contrast, category "3" presents a more significant challenge, with PREC and REC scores dropping to 0.52 and 0.37, respectively. This suggests that the model struggles to identify true instances within category "3," leaving ample room for improvement in its predictive capabilities for this category. We observe a generally high predictive proficiency when considering the macro average PREC and REC, both at 0.76, along with weighted averages of 0.95. However, these scores are slightly affected by the weaker performances in categories "2" and "3." Despite these challenges, the model maintains an overall ACC of 94.91%, signifying its competence. Nevertheless, there is potential for enhancement in distinguishing the nuances between categories "2" and "3," which could further elevate its efficiency.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	4696
1	0.99	0.99	0.99	4283
2	0.60	0.70	0.64	526
3	0.52	0.37	0.43	420
accuracy			0.95	9925
macro avg	0.77	0.76	0.76	9925
weighted avg	0.95	0.95	0.95	9925

Figure 4.97. Performance Evaluation of LR

XGBoost

Utilizing XGBoost, an advanced implementation of GB, our analysis for classifying news into 'fake', 'true', 'half true', and 'barely true' categories yielded remarkable outcomes through our cross-validation process. The algorithm, known for its efficiency and effectiveness in handling various types of data, has demonstrated its prowess in the nuanced field of FND.

Initiating with fold one, the model manifested its predictive capabilities with an impressive ACC of 95.48%. This high level of PREC was consistently maintained in fold two, albeit with a slight decrease, achieving an ACC of 95.38%. Fold three observed a marginal increase, reinforcing the model's adeptness with an ACC of 95.47%. Progressing to fold four, the model reached its peak ACC of 95.51%, the highest among the five folds, showcasing the potential of XGBoost in understanding and categorizing complex, nuanced datasets.

The final fold concluded with a test ACC of 95.19%, a slight retraction from the previous fold but still significantly high, underscoring the model's robustness. The overall ACC after cross-validation, aligning with the fifth fold's result, was 95.19%. This consistent performance across different subsets of the data illustrates the model's stability and its ability to generalize well, which is critical for the reliable classification of information ACC.

In our assessment of the XGBoost model using a CM, we observe exceptional PREC and REC for the "fake" and "true" categories (Figure 4.98). The model demonstrates near-perfect discrimination of instances within these classes, as indicated by the high counts of TP: 4664 out of 4696 for "fake" and 4273 out of 4283 for "true." Additionally, the PREC values

approach 1.00, underscoring the model's robust performance in accurately classifying the majority classes.

Conversely, the model's performance diminishes when dealing with the "half true" and "barely true" categories, revealing its limited effectiveness in correctly categorizing instances in these groups. Specifically, the "half true" category exhibits a REC of 0.64, while the "barely true" category achieves a REC of 0.42. These values are calculated based on the number of correctly identified instances: 336 out of 526 for "half true" and 175 out of 420 for "barely true."

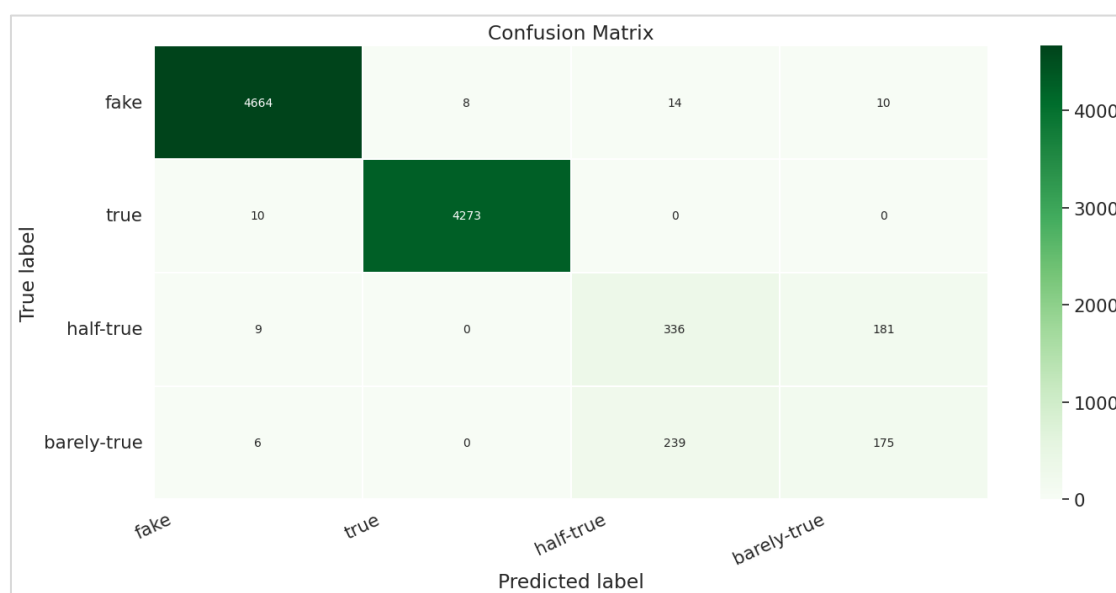


Figure 4.98. Evaluation Matrix for XGBoost

The XGBoost model exhibits exceptional competence when it comes to classifying instances into the “0” and “1” categories, as evidenced by PREC and REC values that are nearly perfect, surpassing 99% (Figure 4.99). The F1-Ss, which represent the harmonic mean of PREC and REC, reiterate this remarkable level of performance, with values of 0.99 and 1.00 for categories “0” and “1,” respectively. This suggests a well-balanced classification capability with minimal FP and FN. However, as we shift our focus to categories “2” and “3,” we observe a decline in performance. The REC for category “2” stands at 0.64, indicating that only 64% of the actual “2” instances have been correctly identified, leaving a significant portion misclassified. The PREC for this category is 0.57, meaning that out of the instances classified as “2,” 57% were correct classifications. On the other hand, the “3” category exhibits the lowest performance metrics across PREC, REC, and F1-S, with

respective values of 0.48, 0.42, and 0.45. These values suggest a moderate classification performance, with notable instances either being missed or incorrectly classified as category “3.” Despite the relatively weaker performance in categories “2” and “3,” the overall ACC of the model remains robust at 95.19%. The PREC, REC, and F1-S macro averages hover around 0.76, indicating a generally balanced performance when considering all categories. However, there is room for improvement in the less-performing categories to achieve a more uniform performance across all classes.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	4696
1	1.00	1.00	1.00	4283
2	0.57	0.64	0.60	526
3	0.48	0.42	0.45	420
accuracy			0.95	9925
macro avg	0.76	0.76	0.76	9925
weighted avg	0.95	0.95	0.95	9925

Figure 4.99. Performance Evaluation of XGBoost

Decision tree

In the intricate endeavour of classifying news into tiers of veracity—'fake', 'true', 'half true', and 'barely true'—the DT classifier was rigorously evaluated across a five-fold cross-validation sequence. This classic ML algorithm, celebrated for its interpretability and simplicity, was subjected to a rigorous analysis to ascertain its efficacy in distinguishing the gradations of truth within news statements.

The initial fold commenced with an encouraging ACC of 94.95%, setting a robust benchmark for the classifier's performance. Slightly advancing this measure, fold two recorded a marginally higher ACC of 95.00%, hinting at the model's consistent classification prowess. However, fold three observed a dip, with ACC descending to 94.49%, the lowest in our series, possibly indicating variability in the model's performance with different data subsets. Fold four saw a nominal recovery in ACC to 94.57%, suggesting a return towards the initial performance levels. The concluding fold five demonstrated an improved accuracy of 94.87%, indicative of the model's resilient classification capability.

The comprehensive test accuracy, calculated post the cross-validation process, aligned with the fifth fold, affirming a final figure of 94.87%.

Upon analyzing the CM for the DT classifier, it becomes evident that the model demonstrates a high level of competence when it comes to classifying the “fake” and “true” categories, correctly identifying a significant majority of instances (Figure 4.100). Specifically, the PREC scores for both “fake” and “true” are noteworthy, indicating a high level of ACC in the model's predictions for these categories. Additionally, the REC values for these classes are impressive, signifying that a substantial percentage of all actual instances belonging to these categories have been correctly classified.

However, the model's performance becomes somewhat less satisfactory when we shift our focus to the “half true” and “barely true” categories. For the “half true” category, the PREC value is derived from 315 correctly predicted instances but is impacted by 242 incorrect predictions (comprising 197 FP in “barely true” and a combined 14 in the other categories), resulting in a suboptimal PREC score. The REC metric reveals that out of 526 actual “half true” instances, 211 were misclassified, indicating room for improvement in the model's sensitivity to this category. As for the “barely true” category, the classifier correctly identified 181 instances, but a significant number of misclassifications, totalling 239 (231 as “half true” and a combined 8 in the other categories), were observed.

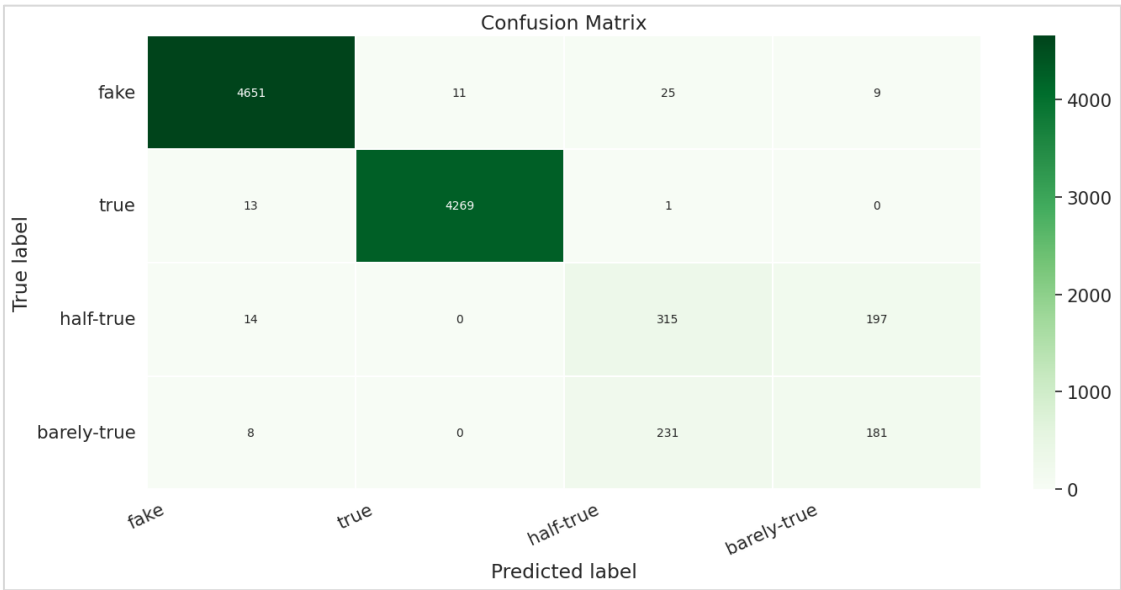


Figure 4.100. Evaluation Matrix for DT

When assessing the performance of the DT classifier using the provided CM, it's evident that the model excels in classifying instances into the "fake" (0) and "true" (1) categories, as indicated by the exceptionally high PREC and REC scores that approach 100% for these classes (Figure 4.101). This highlights the model's ability to accurately identify and categorize nearly all instances belonging to these straightforward "fake" and "true" categories, showcasing its robustness in distinguishing between these clearly defined data points. However, a closer examination of the performance metrics related to the "half true" (2) and "barely true" (3) categories reveals that the model encounters challenges when classifying these more nuanced categories accurately. The F1-Ss for categories 2 and 3 are 0.57 and 0.45, respectively, indicating that the harmonic mean of PREC and REC for these categories is not as high as that for the "fake" and "true" categories. The relatively lower PREC and REC values for categories 2 and 3 underscore the model's limited capability to classify instances within these categories effectively. To provide more specific details, we observe a PREC of 0.55 and a REC of 0.60 for the "half true" category, as well as a PREC of 0.47 coupled with a REC of 0.43 for the "barely true" category, indicating a moderate level of performance in these areas. Nonetheless, the overall ACC of the model remains impressive, hovering at 94.87%.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	4696
1	1.00	1.00	1.00	4283
2	0.55	0.60	0.57	526
3	0.47	0.43	0.45	420
accuracy			0.95	9925
macro avg	0.75	0.75	0.75	9925
weighted avg	0.95	0.95	0.95	9925

Figure 4.101. Performance Evaluation of DT

Multinomial naive Bayes

The MNB model, a statistical classifier that applies Bayes' theorem with the assumption of independence between predictors, was leveraged to classify news articles into 'fake', 'true', 'half true', and 'barely true' categories. The results gleaned from a stringent five-fold cross-validation regimen offer insights into its aptitude for this nuanced task.

In the inaugural fold, the model demonstrated a commendable level of ACC at 89.24%, providing a solid base to ascertain the model's performance. Fold two exhibited an uptick in ACC, climbing to 89.90%, suggesting the model's capability to effectively capture the distinguishing features of the dataset for classification purposes.

A discernible dip occurred in fold three, with the ACC decreasing to 87.41%. This could indicate the model's variability in performance with different data distributions, a common occurrence in probabilistic models sensitive to the input data's characteristics. However, fold four recovered in ACC to 89.07%, pointing towards the model's resilience and adaptability.

The final fold concluded the sequence with the highest ACC observed, at 91.08%. This peak in the model's performance signifies the potential for MNB to deliver strong results, particularly when the feature space is representative of the data's underlying patterns. The overall test ACC, recorded after the cross-validation process, echoed the high performance of the final fold at 91.08%.

Upon analyzing the CM for the MNB classifier (Figure 4.102), it becomes evident that the model exhibits significant discriminative capabilities when distinguishing between the 'fake' and 'true' classes. This is substantiated by a substantial number of instances correctly classified, as evidenced by the high counts along the diagonal elements corresponding to these classes. However, it's worth noting that many instances have also been misclassified, represented by the off-diagonal elements in the CM. Specifically, there is a notable occurrence of misclassifications between the 'true' class being erroneously labelled as 'fake' and vice versa.

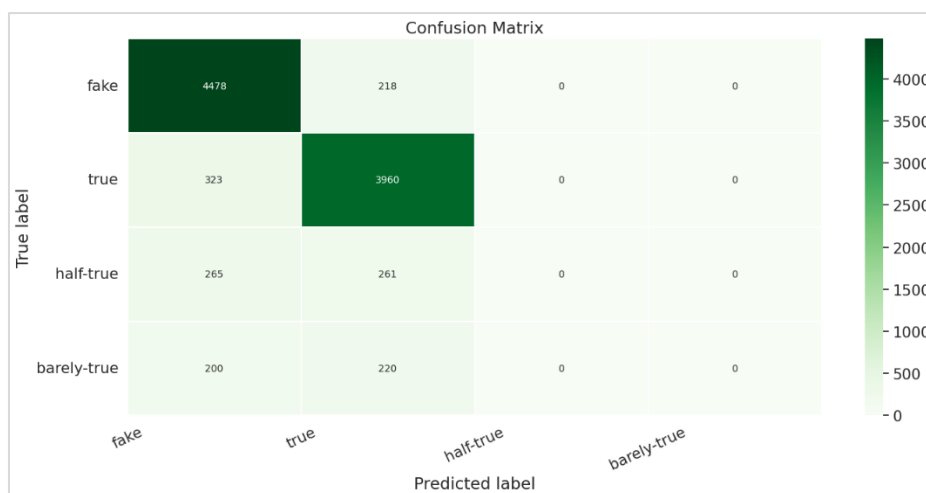


Figure 4.102. Evaluation Matrix for MNB

The results obtained from the CR after applying the MNB model to the dataset show that the model excels in identifying the 'fake' and 'true' categories. This is indicated by these classifications' high PREC and REC values, exceeding 0.85 (Figure 4.103). Additionally, the F1-Ss for these categories are commendable, approaching 0.90, which suggests a well-balanced trade-off between PREC and REC. However, a significant limitation of the model is apparent in its performance on the 'half true' and 'barely true' classes. The model fails to correctly classify instances in these categories, resulting in PREC, REC, and F1-S metrics displaying zero values for these classes. This substantial deficiency highlights that the model simplifies the problem to a binary classification task, neglecting the nuanced distinctions required for accurate categorization into the 'half true' and 'barely true' categories. Although the model achieves a reasonably high ACC of 0.85, the macro average metrics reveal its limitations. The values are less than half of the maximum possible value of 1, indicating that, on average, the model's performance across all classes is not robust. The notable difference between the weighted and macro average results underscores the model's bias towards the better-represented classes.

	precision	recall	f1-score	support
0	0.85	0.95	0.90	4696
1	0.85	0.92	0.89	4283
2	0.00	0.00	0.00	526
3	0.00	0.00	0.00	420
accuracy			0.85	9925
macro avg	0.43	0.47	0.45	9925
weighted avg	0.77	0.85	0.81	9925

Figure 4.103. Performance Evaluation of MNB

SVM-RBF

The SVM with RBF kernel has been rigorously applied to classify news into 'fake', 'true', 'half true', and 'barely true'. The SVM-RBF is a non-linear classifier that excels in finding complex patterns in data by projecting it into higher-dimensional spaces where the classes are more easily separable. This capability was tested through a five-fold cross-validation approach to ensure the robustness and reliability of the results.

In fold one, the SVM-RBF classifier commenced with a test ACC of 87.89%, providing a solid baseline for its performance in our task. Fold two showed an increase in ACC, achieving a more impressive 88.78%, reflecting the classifier's capability to effectively

generalize from the training data. A slight decline in performance was observed in fold three, with the ACC dipping to 87.77%. This may reflect the variability inherent in the distribution of news categories within that particular subset of data. Nevertheless, fold four saw a resurgence to a higher ACC of 88.92%, indicating the model's resilience and adaptability to different data segments.

The culmination of our cross-validation, fold five, achieved the highest ACC of the sequence at 89.27%, underscoring the SVM-RBF model's potential in effectively discerning the veracity of news. The overall test ACC, harmonious with the result of fold five, solidified at 89.27%, affirming the consistency and predictive strength of the classifier.

In this CM analysis resulting from utilising an SVM with RBF kernel, we observe a competent performance in the 'fake' and 'true' categories. This is evident from the high PREC and REC values associated with these classes (Figure 4.104). Specifically, the 'fake' category exhibits a commendable PREC value, exceeding 0.90, which highlights the model's ability to identify a significant proportion of instances in this category correctly. Similarly, the 'true' category demonstrates proficient identification, albeit with a slightly lower PREC. However, it compensates with a substantial REC figure, indicating a robust identification of actual positives.

Conversely, when we turn our attention to the 'half true' and 'barely true' categories, a decline in performance becomes apparent, characterized by reduced PREC and REC values. This decline signifies the model's difficulty distinguishing the subtler nuances that differentiate these categories. Particularly in the 'barely true' category, the model appears to misclassify a considerable number of instances as belonging to other categories, as evidenced by the lower REC figure. Regarding the F1-S, which represents the harmonic mean of PREC and REC, we observe strong scores for the 'fake' and 'true' categories, underscoring the model's proficiency in these areas. Conversely, the 'half true' and 'barely true' categories register lower scores, indicating a less harmonious balance between PREC and REC for these classes.

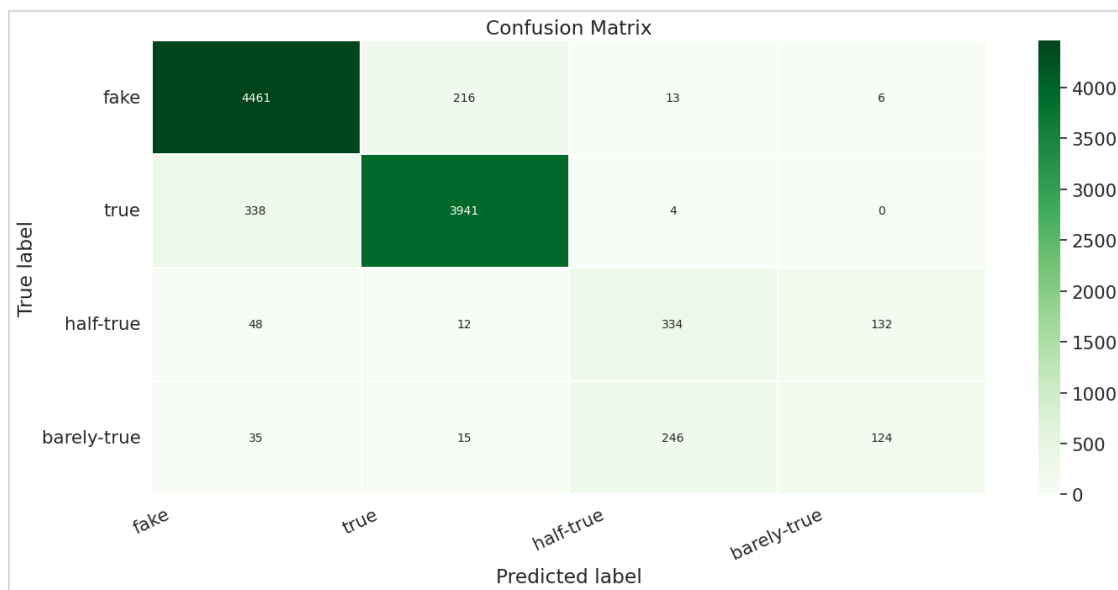


Figure 4.104. Evaluation Matrix for SVM-RBF

Upon analyzing the CR for the SVM with RBF kernel, it becomes evident that the model exhibits a robust performance when distinguishing between the primary classes denoted as '0' and '1'. PREC and REC values for these classes exceed 0.90 (Figure 4.105), signifying a high rate of correct predictions and a substantial ability to identify TP within these categories. This strong performance establishes the model as a competent classifier for the predominant classes. However, the report also highlights a significant drop in the model's effectiveness when dealing with the minor classes labelled '2' and '3'. For class '2', although the F1-S falls below 0.60, it indicates a moderate balance between PREC and REC, with both values surpassing 0.55. This suggests that while there is room for improvement, the model still makes substantial correct predictions in this category.

Conversely, the '3' category demonstrates a substantial underperformance, with the lowest F1-S among the groups. This indicates a relative deficiency in balancing PREC and REC for this category. The REC value of 0.30 implies that a significant portion of actual positives is being misclassified, highlighting the model's reduced effectiveness in recognizing instances of this class. This underscores the need for critical improvements in this aspect. Examining the macro average and weighted average values is essential to gain a broader perspective on the model's overall performance. These values suggest that the model is proficient as a classifier, although it exhibits notable shortcomings in accurately distinguishing the subtler variations present in the less represented classes.

	precision	recall	f1-score	support
0	0.91	0.95	0.93	4696
1	0.94	0.92	0.93	4283
2	0.56	0.63	0.59	526
3	0.47	0.30	0.36	420
accuracy			0.89	9925
macro avg	0.72	0.70	0.71	9925
weighted avg	0.89	0.89	0.89	9925

Figure 4.105. Performance Evaluation of SVM-RBF

Linear SVM

The Linear-SVM classifier, renowned for its efficiency in high-dimensional spaces, was deployed to distinguish between varying levels of truth in news articles, categorized as 'fake', 'true', 'half true', and 'barely true'. The results from a five-fold cross-validation framework offer a comprehensive view of its performance in this multi-class classification task.

Commencing with fold one, the Linear-SVM achieved a test ACC of 89.24%, indicating a promising start to the classification endeavour. Fold two witnessed an improvement, with the ACC climbing to 89.90%, suggesting that the Linear-SVM is adept at handling the nuances of the dataset. However, fold three presented a noticeable decrease in ACC, dropping to 87.41%. This reduction might reflect the model's sensitivity to the particular distribution of the data in that fold, highlighting areas where the model's linear decision boundary could be less effective.

The ACC rebounded in fold four to 89.07%, demonstrating the classifier's recovery capacity and overall stable performance across different data segments. The final fold presented the most striking result, with the ACC soaring to 91.08%, the highest among the folds. This peak illustrates the potential of the Linear-SVM when the model and data are well-aligned.

The aggregate performance across all folds culminated in an overall test ACC of 91.08%, mirroring the result from the final fold and underscoring the Linear-SVM's robustness.

When examining the CM of the Linear SVM model, we can observe that the model demonstrates significant proficiency in accurately classifying instances belonging to the

'fake' and 'true' categories. Most predictions align with the diagonal of the matrix, indicating correct classifications (Figure 4.106). Specifically, the TP rates for these categories are notably high, indicating the model's ability to correctly label observations in the 'fake' and 'true' categories. As we transition to the 'half true' and 'barely true' categories, there is a noticeable decrease in classification PREC. However, it's important to highlight that many predictions for these categories still fall within the correct diagonal cells, suggesting a moderate level of accurate classification.

Nevertheless, a noteworthy number of instances are misclassified into other categories, particularly involving confusion with the 'fake' and 'true' categories. This suggests a potential overlap or proximity in the feature space that the model struggles to distinguish distinctly. Upon closer examination of the off-diagonal elements representing false classifications, it becomes apparent that the false positive and FN rates are higher for the minor classes than the major classes.

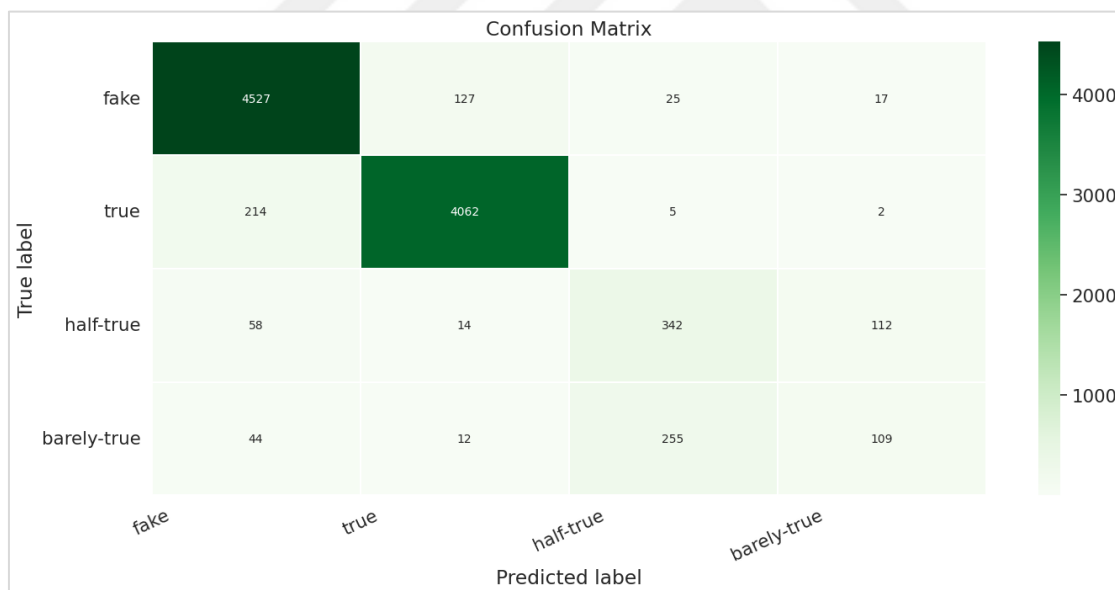


Figure 4.106. Evaluation Matrix for Linear SVM

When examining the CR for the Linear SVM model, it becomes apparent that it excels in differentiating between the '0' and '1' categories, as evidenced by the high PREC and REC values associated with these groups (Figure 4.107). Notably, the model maintains a high level of ACC, with a score of 0.91, indicating that many instances have been correctly classified. However, upon closer inspection of the '2' and '3' classes, it becomes evident that the model's performance metrics decrease, indicating challenges in accurately classifying

these groups. The F1-Ss for these categories, 0.59 for '2' and 0.33 for '3', are notably lower than those for '0' and '1', signifying a compromised harmonic mean of PREC and REC. Specifically, the REC for class '3' is only 0.26, highlighting the model's difficulty in identifying a substantial portion of actual positive instances for this class. Despite the reduced metrics for classes 2 and 3, it's important to acknowledge the decent REC score of 0.65 for class 2. This suggests that the model possesses a moderate ability to capture TP in this category. It implies that while the model struggles to maintain a high PREC in class '2', it prioritises achieving a higher REC, indicating an area with potential for further investigation and optimization during the model fine-tuning phase. The macro average PREC, REC, and F1-S, all around 0.71, provide an overview of the model's overall performance. This suggests that the model can reasonably differentiate between the various classes but also highlights the potential for improvement. Strategies such as refining the feature space or incorporating more relevant data points for the underperforming classes could enhance the model's classification robustness, ultimately improving the classification metrics for the '2' and '3' groups and elevating the overall model performance.

	precision	recall	f1-score	support
0	0.93	0.96	0.95	4696
1	0.96	0.95	0.96	4283
2	0.55	0.65	0.59	526
3	0.45	0.26	0.33	420
accuracy			0.91	9925
macro avg	0.72	0.71	0.71	9925
weighted avg	0.91	0.91	0.91	9925

Figure 4.107. Performance Evaluation of Linear SVM

Discussion

In our comprehensive evaluation using dataset 2, adding the "barely-true" class introduced a layer of complexity to the FND task (Table 4.7). This development resulted in diverse performances across the ML models, especially when dealing with the newly introduced categories.

The RF model achieved an ACC of 88.25%. While it demonstrated strong results for the 'Fake' and 'Real' classes, with PREC and REC rates ranging from 90% to 94% and 94% to

91%, respectively, it faced challenges with the new classes, namely 'half-true' and 'barely-true'. The model exhibited lower performance in these cases, with a PREC of 54% and REC of 59% for the 'half-true' class and a disappointing 45% PREC and 30% REC for the 'barely-true' class.

GB displayed a significant performance improvement, achieving an ACC of 94.89%. It achieved near-perfect scores for the traditional 'Fake' and 'Real' classes, but its performance was less optimal for the 'half-true' and 'barely-true' categories, with PREC-REC figures ranging from 55% to 76% and 49% to 29%, respectively. The model excelled in binary classification but faced limitations when dealing with finer distinctions of truth.

LR achieved a similar ACC of 94.91%, with equally impressive metrics for the 'Fake' and 'Real' classes. Interestingly, the model showed a slight improvement in performance for the newer classes, achieving a PREC of 60% and REC of 70% for the 'half-true' category and a PREC of 52% and REC of 37% for the 'barely-true' class.

XGBoost outperformed other models with an ACC of 95.19%. It performed exceptionally in the 'Fake' and 'Real' classes, achieving almost perfect scores. However, it recorded moderate results in the newly introduced categories. For 'half-true,' it achieved a PREC of 57% and REC of 64%; for 'barely-true,' the metrics were 48% and 42%, respectively.

The DT model closely followed XGBoost, with an ACC of 94.87%. It maintained consistent performance across all classes, achieving near-perfection in the 'Fake' and 'Real' classes but displaying moderate results for the 'half-true' and 'barely-true' classes.

Both SVM models, Linear and RBF, reported accuracies of 91.08% and 89.26%, respectively. They demonstrated strong results for the 'Fake' and 'Real' categories but exhibited inconsistent performance for the new classes. For instance, the Linear SVM model achieved a PREC of 55% and REC of 65% for the 'half-true' category and a PREC of 45% and REC of 26% for the 'barely-true' category.

MNB presented unique results in the table, with accuracies for the 'Fake' and 'Real' classes, featuring PREC and REC rates ranging from 85% to 95% and 85% to 92%, respectively.

However, it struggled to provide meaningful predictions for the new classes, resulting in 0% PREC, REC, and F-score for both the 'half-true' and 'barely-true' categories.

Table 4.7. ML classifiers result on dataset2

Classifier	ACC	Class	PREC	REC	F-Score
RF	88.25%	Fake	90%	94%	92%
		Real	94%	91%	92%
		Half-True	54%	59%	56%
		Barely-True	45%	30%	36%
GB	94.89%	Fake	100%	98%	99%
		Real	99%	100%	100%
		Half-True	55%	76%	64%
		Barely-True	49%	29%	36%
XGBoost	95.19%	Fake	99%	99%	99%
		Real	100%	100%	100%
		Half-True	57%	64%	60%
		Barely-True	48%	42%	45%
Linear SVM	91.08%	Fake	93%	96%	95%
		Real	96%	95%	96%
		Half-True	55%	65%	59%
		Barely-True	45%	26%	33%
RBF SVM	89.26%	Fake	91%	95%	93%
		Real	94%	92%	93%
		Half-True	56%	63%	59%
		Barely-True	47%	30%	36%
LR	94.91%	Fake	99%	99%	99%
		Real	99%	99%	99%
		Half-True	60%	70%	64%
		Barely-True	52%	37%	43%
DT	94.87%	Fake	99%	99%	99%
		Real	100%	100%	100%
		Half-True	55%	60%	57%
		Barely-True	47%	43%	45%
MNB	85.01%	Fake	85%	95%	90%
		Real	85%	92%	89%
		Half-True	0%	0%	0%
		Barely-True	0%	0%	0%

4.6.2. Deep learning classifiers results

LSTM

In this series of experiments, the LSTM model underwent training and validation using a five-fold approach, demonstrating its ability to progressively learn and accurately classify data points with a high degree of PREC. This is evident from the ACC metric observed across each epoch (Figure 4.108). The training process was characterized by a substantial

decrease in loss values, signifying the model's increasing proficiency in making correct predictions as it iterated over the training dataset multiple times. In the initial epochs of each fold, a noticeable reduction in training loss and validation loss was observed, confirming the model's effectiveness in learning. In the first fold, the model achieved an impressive validation ACC of 99.92% in the first epoch, maintaining this high standard throughout subsequent epochs and ultimately resulting in a test ACC of 95.44%. In the following folds, the model consistently improved its validation ACC, reaching a peak of 100% in folds 2 and 4. Notably, by the end of training in fold 2, the model had already attained a maximum validation ACC of 100%, highlighting its optimized performance. After the last epoch in each fold, the test results reaffirmed the model's effectiveness, with test accuracies consistently exceeding 95%. This demonstrated the model's strong ability to generalize the learned patterns to unseen data.

Furthermore, as we progressed through the later folds, a consistent trend of increasing ACC and decreasing loss was observed in both the training and validation phases, culminating in peak performance in fold 5, where the model achieved an impressive test ACC of 98.65%. This indicates that the model is well-tuned and skilled at recognizing and accurately predicting patterns in the data, underscoring its robustness and reliability in classification tasks. However, despite these promising results, it is important to remain cautious about the potential for overfitting. This concern arises due to the repeated high validation accuracies and the notable contrast between training and validation loss, particularly in the later epochs of folds 2, 4, and 5.



Figure 4.108. Accuracy and loss metrics for training and validation of LSTM

Analyzing the outcomes from the CM of the LSTM model, it becomes evident that the model possesses an exceptional capability to accurately categorize instances into the 'fake' and 'true'

classes. This is exemplified by the substantial number of TP: 4695 for 'fake' and 4283 for 'true', with no instances of FN or FP recorded for the 'true' category (Figure 4.109). This underscores the model's outstanding classification proficiency for these two categories. Shifting our attention to the 'half true' and 'barely true' categories, we observe a higher misclassification rate, although the overall results remain commendable. The 'half true' category saw 490 correct predictions while also experiencing a notable number of instances (97) being erroneously classified as 'barely true'. Conversely, the 'barely true' category exhibited strong performance with 323 accurate classifications, albeit with 29 instances mistakenly identified as 'half true'.

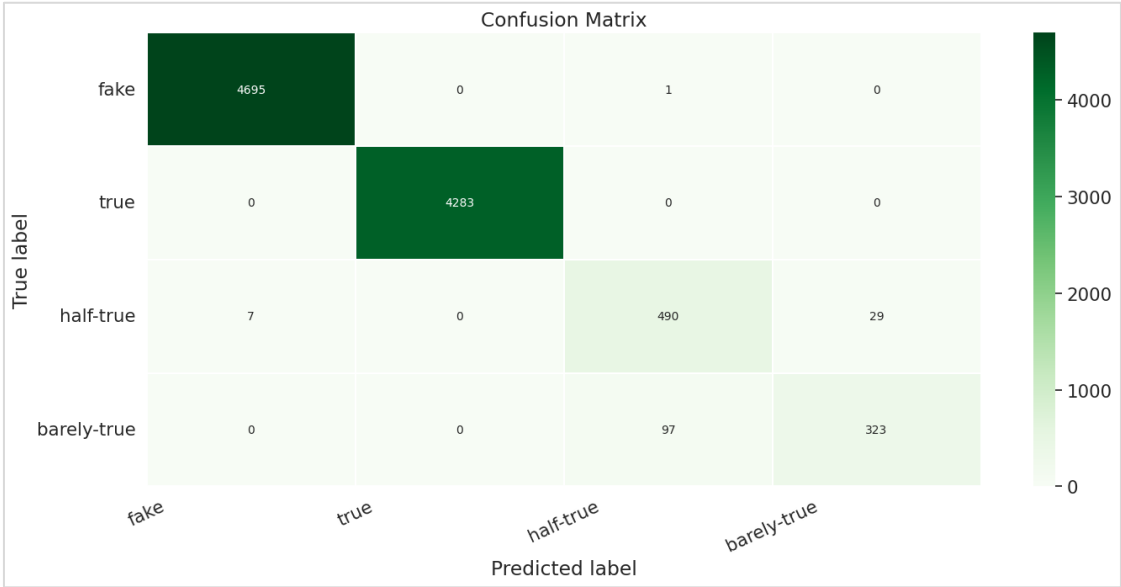


Figure 4.109. Evaluation Matrix for LSTM

The CR for the LSTM model provides insights into its remarkable ACC and PREC when it comes to categorizing data points into the four designated groups (Figure 4.110). Specifically, when examining the PREC metric, which gauges the model's ability to identify TP among all instances it identifies as positives accurately, we find perfect scores of 1.00 for both categories 0 and 1. These scores indicate that the model performs flawlessly in these categories, with no FP recorded. This trend is also reflected in the REC metric, which calculates the fraction of actual positives correctly identified, highlighting the LSTM's ability to distinguish these categories. However, as we delve into categories 2 and 3, we observe slightly lower PREC values of 0.93 and 0.77, respectively, accompanied by REC values of 0.83 and 0.92 for the same categories. While these scores are still relatively high, they do not match the impeccable performance seen in categories 0 and 1, suggesting a

moderate degree of misclassification and room for improvement in the model's predictive ACC for these categories. The F1-Ss, which represent a harmonic mean of PREC and REC, exhibit a similar trend, with excellent results in the initial two categories and slightly lower yet robust scores in the latter.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4702
1	1.00	1.00	1.00	4283
2	0.93	0.83	0.88	588
3	0.77	0.92	0.84	352
accuracy			0.99	9925
macro avg	0.93	0.94	0.93	9925
weighted avg	0.99	0.99	0.99	9925

Figure 4.110. Performance Evaluation of LSTM

CNN-LSTM

Using 5-fold cross-validation to train and validate the CNN-LSTM model, we observe distinctive trajectories for both ACC and loss throughout 5 epochs per fold (Figure 4.111). The loss component is computed using sparse categorical cross-entropy, while ACC provides a straightforward measure of correct predictions out of total predictions.

In the initial fold, the model initiated with a training loss of 0.6926, which saw a significant reduction over subsequent epochs, culminating at 0.1304 by the fifth epoch. Concurrently, the validation loss also substantially declined over epochs, indicative of a well-optimized model. The training ACC began at 64.86% and displayed steady growth, reaching its peak at 94.24% in the final epoch, underscoring the model's capacity to learn effectively. Notably, the validation ACC showed a remarkable leap, commencing at a promising 97.83% and reaching an impressive 99.67% in the last epoch. The test results for this fold revealed a loss of 0.1013 and an ACC of 94.93%. As we progress through subsequent folds (2 to 5), we observe a consistent trend of diminishing loss and increasing ACC over the epochs, albeit with periodic fluctuations. Both training and validation losses displayed oscillatory patterns, illustrating the model's process of fine-tuning and balancing between underfitting and overfitting. The final fold demonstrated the model's adept learning capability, with training loss and ACC stabilizing at 0.0819 and 96.27% at the end of the epoch sessions. This fold

marked substantial progress from the initial state, indicating commendable predictive power. Upon examining the validation metrics across all folds, it becomes evident that the model consistently maintained high ACC, typically exceeding 99%, while simultaneously keeping validation loss low, a sign of a well-generalized model. Furthermore, the test accuracies across all folds consistently met a high standard, remaining above 94% and achieving a peak of 96.06% in the last fold.



Figure 4.111. Accuracy and loss metrics for training and validation of CNN-LSTM

When we examine the CM of the CNN-LSTM model, we gain insights into its classification performance across four distinct categories: 'fake,' 'true,' 'half true,' and 'barely true' (Figure 4.112). Let's begin by focusing on the 'fake' and 'true' categories, where the model demonstrated remarkable PREC. It accurately classified 4678 out of 4682 instances as 'fake' and 4279 out of 4280 as 'true.' These results indicate a high level of ACC in distinguishing between 'fake' and 'true' statements, with only minor errors. However, a more intricate picture emerges when we consider the classification of the 'half true' and 'barely true' categories. In the case of 'half true,' the model correctly identified 509 instances but made 352 misclassifications as 'barely true,' and 2 instances were incorrectly labelled as 'fake.' In the 'barely true' category, the model accurately identified 68 instances but also exhibited a notable number of misclassifications, including 15 incorrectly categorised as 'fake' and 17 as 'half true.' This portion of the CM reveals a greater susceptibility to errors, highlighting a potential area for model refinement to enhance its ability to distinguish between nuanced classifications.

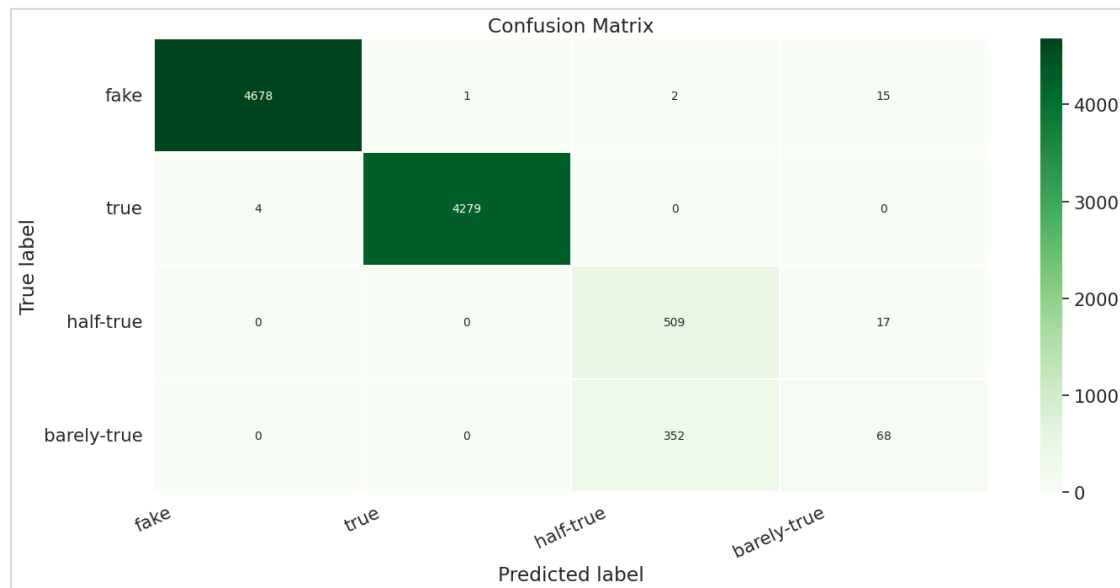


Figure 4.112. Evaluation Matrix for CNN-LSTM

Analyzing the CR for the CNN-LSTM model provides valuable insights into its performance across various metrics, including PREC, REC, and F1-S (Figure 4.113). The data reveals that the model has achieved near-perfection in identifying categories 0 (representing 'fake') and 1 ('true'). It boasts a PREC, REC, and F1-S of 1.00 for both categories, indicating an exceptionally successful classification with a remarkable alignment between predicted and actual values. However, a different pattern emerges when we delve into categories 2 ('half true') and 3 ('barely true'). Category 2 exhibits a commendable PREC score of 0.97, but its REC is notably lower at 0.59, resulting in an F1-S of 0.73. This suggests that while the model correctly predicts instances as Category 2, it misses a significant portion of actual Category 2 instances. There is room for improvement in boosting the REC metric for this category. The most significant area for improvement is observed in category 3. Despite a reasonable REC of 0.68, the PREC is considerably low at 0.16, leading to a diminished F1-S of 0.26. This indicates that many instances are erroneously classified as category 3, emphasizing the need for further optimization to enhance PREC. Despite these specific challenges, the overall ACC of the model stands at a robust 96.06%, with high aggregate scores in the weighted average metrics. This illustrates that the model performs exceptionally well on a large portion of the dataset, showcasing its overall effectiveness.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4682
1	1.00	1.00	1.00	4280
2	0.97	0.59	0.73	863
3	0.16	0.68	0.26	100
accuracy			0.96	9925
macro avg	0.78	0.82	0.75	9925
weighted avg	0.99	0.96	0.97	9925

Figure 4.113. Performance Evaluation of CNN-LSTM

CNN

In a five-fold series of experiments, we closely monitored the training and validation of ACC and loss curves for the CNN model throughout five epochs in each fold, capturing essential metrics at each stage (Figure 4.114). In the initial fold, the model commenced with a relatively high loss of 6.1484, which consistently decreased as the epochs progressed, ultimately stabilizing at 0.7578 by the fifth epoch. This substantial reduction in loss reflects the model's learning process as it gradually minimizes errors during training. However, despite the decrease in loss, the final test ACC settled just above 48%. Moving on to the second fold, we observed an encouraging upward trend in ACC, starting at around 51% and reaching a promising 58.81% over the five epochs. A continuous reduction in loss accompanied this improvement. It indicates a model slowly but steadily enhancing its predictive capabilities as it undergoes fine-tuning in each epoch. By the third fold, the model demonstrated a more solid understanding, achieving a test ACC of 59.80%. However, it grappled with fluctuations in validation loss and ACC throughout the training epochs, suggesting challenges in optimizing the loss function during this particular round of training. As we entered the fourth fold, there was a noticeable roller-coaster pattern in validation ACC and loss. The model oscillated before settling at a less satisfying ACC of 56.87%. This scenario suggests that the model might have started overfitting to the training data, as indicated by the increase in validation loss in the later epochs. Finally, in the fifth fold, the model faced the task of maintaining diminishing loss while striving to improve ACC. Unfortunately, the last epoch witnessed a slight dip in ACC. The test ACC managed to slightly surpass the 61% mark, providing a glimpse of the potential for further optimizations in the model.

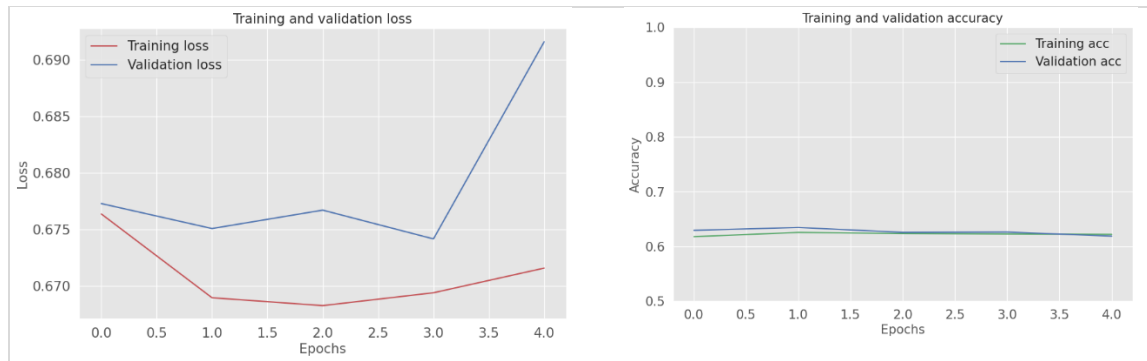


Figure 4.114. Accuracy and loss metrics for training and validation of CNN

Examining the CNN's results through the CM provides a clear understanding of the model's classification performance across four distinct categories: "fake," "true," "half true," and "barely true" (Figure 4.115). Let's start with the "fake" category, where the model successfully identified 3002 instances but also had a notable number of FP, with 1634 instances misclassified. This indicates a tendency for the model to confuse "fake" with "true occasionally." Similarly, within the "true" category, the model correctly pinpointed 2645 instances but also had 1467 instances wrongly labelled as "fake." This reciprocal confusion between the "fake" and "true" categories suggests a degree of overlap or similarity between these two categories in the model's classification. Moving to the "half true" category, the data reveals a considerable spread in classification ACC, with 494 instances correctly predicted and a noticeable number of misclassifications into the "barely true" category. Notably, the "barely true" category had the most minor data, with only one instance correctly classified. This highlights a significant dataset imbalance and potentially indicates that the model is still developing its ability to recognize patterns in this category.

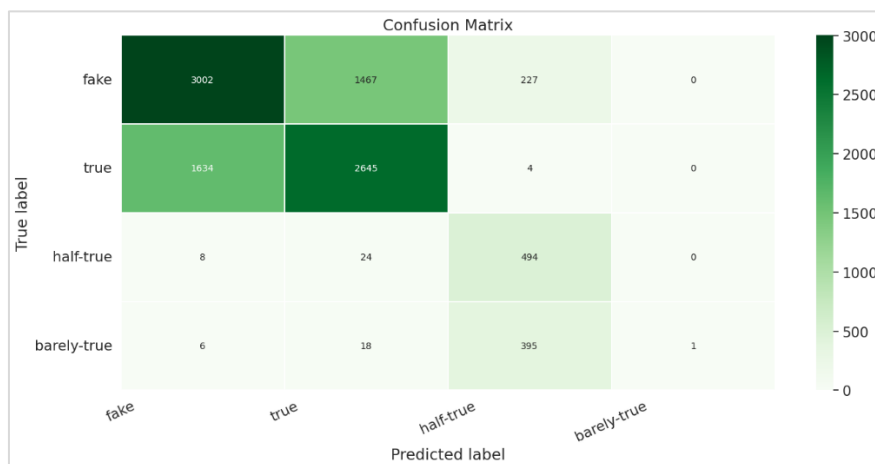


Figure 4.115. Evaluation Matrix for CNN

CNN's CR provides insights into its strengths and weaknesses in accurately predicting the four defined classes: 0, 1, 2, and 3 (Figure 4.116). Let's focus on PREC, which indicates the model's effectiveness in correctly identifying elements belonging to each category. The model excels in PREC in category 2, achieving a commendable score of 0.94. However, it faces challenges in REC, with a rate of only 0.44, resulting in an F1-S of 0.60. This suggests that while the model is good at recognizing TP for this category, it struggles to capture a significant portion of relevant instances. This highlights the need for improvement in discerning the intrinsic features of this category.

In contrast, categories 0 and 1 exhibit a more balanced performance, with PREC values of 0.64 and 0.62 and REC metrics in the mid-sixties percentile. The resulting F1-Ss of 0.64 and 0.63 indicate a reasonably proficient system. However, there is still potential for enhancement through fine-tuning the model to better differentiate between these categories, thus improving PREC and REC values. Category 3 presents a unique situation, characterized by zero PREC but a perfect REC score, resulting in an F1-S of zero. This anomaly is attributed to the limited representation of this category in the test set, with only a single instance. While the model accurately identifies this instance, the zero PREC indicates its tendency to allocate instances to this category falsely. This underscores the need for a more diverse training dataset to achieve this category's balanced and accurate predictive capability. The model achieves an overall ACC rate of 61.88% on the test dataset.

	precision	recall	f1-score	support
0	0.64	0.65	0.64	4650
1	0.62	0.64	0.63	4154
2	0.94	0.44	0.60	1120
3	0.00	1.00	0.00	1
accuracy			0.62	9925
macro avg	0.55	0.68	0.47	9925
weighted avg	0.66	0.62	0.63	9925

Figure 4.116. Performance Evaluation of CNN

LSTM autoencoder

During a thorough examination of the training process of the LSTM autoencoder model, distinct patterns emerge across successive epochs in terms of loss and ACC metrics in both

training and validation phases across the five folds (Figure 4.112). In the initial fold, the model began with a relatively high loss of 0.6992, notably decreasing to 0.6435 by the fifth epoch, indicating a noticeable improvement in its predictive abilities. However, the validation loss exhibited a more fluctuating pattern, ultimately stabilizing at 0.6392. The ACC metric remained relatively constant throughout the epochs, resulting in a test ACC of 43.17%. As the model advanced through subsequent folds, a similar trend of gradually decreasing loss values was observed, signifying a gradual stabilization of the model. Notably, there was a consistent reduction in validation loss across epochs in the third fold, indicating a commendable improvement in the model's ability to generalize, as confirmed by a test ACC of 43.16%. Moving into the fourth and fifth folds, it became evident that loss values, both in training and validation, displayed minor fluctuations but generally exhibited a decreasing trend over time. However, in the final epoch of the fifth fold, there was a slight increase in loss. The ACC metric remained relatively consistent during these phases, hovering around 43.16%. This suggests a plateau in the model's learning capacity, where further improvements are challenging.

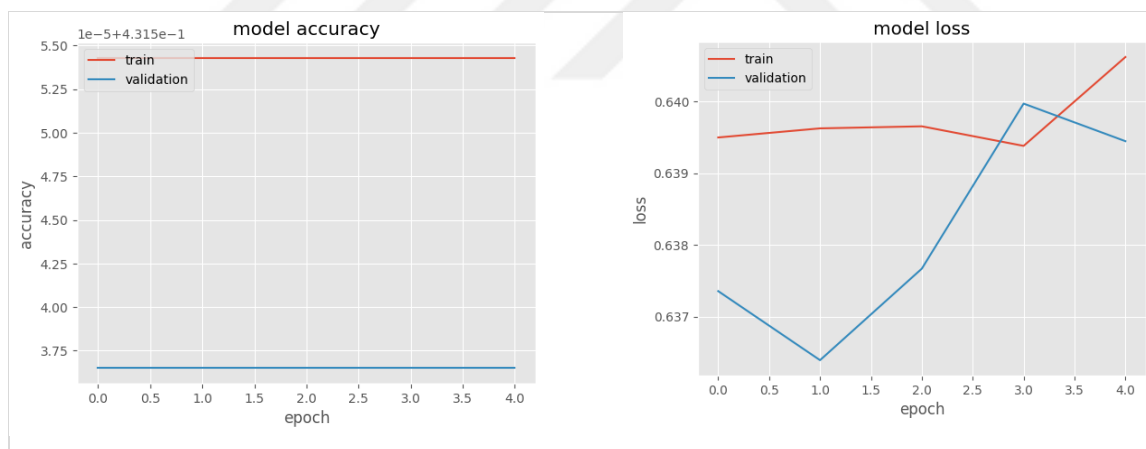


Figure 4.117. Accuracy and loss metrics for training and validation of LSTM Autoencoder

The CM for the LSTM-AutoEncoder model's classification of fake news into different categories provides valuable insights into its performance (Figure 4.118). The matrix reveals that out of the 4,696 instances of fake news, the model correctly classified 3,571 as fake, demonstrating a high TP rate. However, it also misclassified 2,114 fake news articles as true, 957 as half-true, and 2154 as barely true, leading to substantial FP in these categories. The model correctly classified 489 true articles, which indicates a decent TP rate for this class. Surprisingly, it correctly classified 381 half-true articles, suggesting some success in this

category despite its challenges with fake news classification. Unfortunately, the model could not classify any articles as barely true, resulting in a complete misclassification for this class. The matrix illustrates the model's strengths and weaknesses in classifying fake news into various categories, highlighting areas for improvement in its performance.

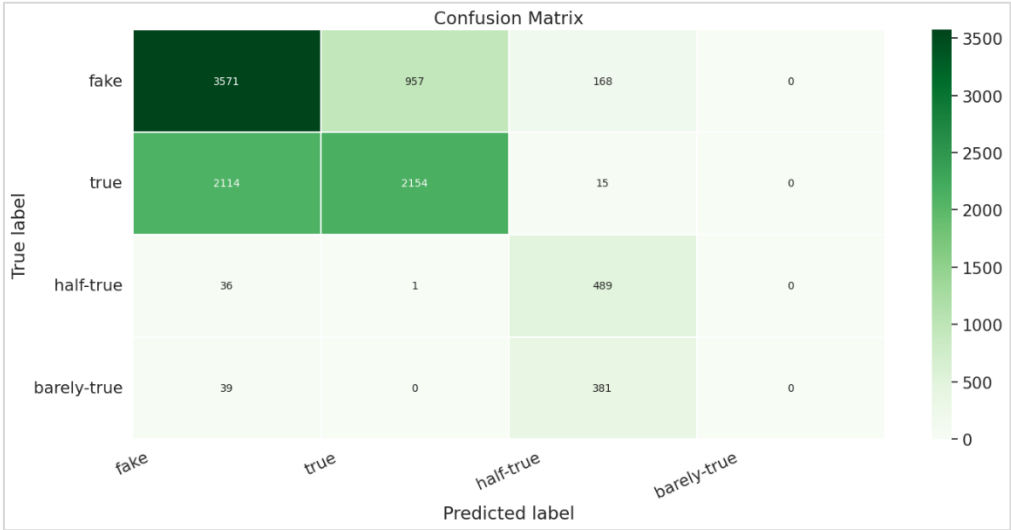


Figure 4.118. Evaluation Matrix for LSTM-AutoEncoder

The CR for the LSTM-AutoEncoder model's performance in categorizing fake news into four classes provides a comprehensive overview of its PREC, REC, and F1-S for each category (Figure 4.119). In the "Fake" category (class 0), the model achieved a PREC of 64%, indicating that when it predicted an article as fake, it was correct 64% of the time. The REC for this class was 65%, indicating that it correctly identified 65% of the fake articles. The F1-S, which balances PREC and REC, stood at 64%, suggesting a reasonable balance between the two metrics.

For the "True" category (class 1), the model demonstrated a PREC of 62%, indicating its ability to classify true articles correctly. The REC for this class was also 64%, signifying its effectiveness in identifying true articles. The F1-S for this category was 63%, indicating a balanced performance between PREC and REC.

In the "Half-True" category (class 2), the model achieved a high PREC of 94%, suggesting a strong ability to identify half-true articles correctly. However, the REC for this class was lower at 44%, indicating that it missed a significant portion of actual half-true articles. The F1-S for this category was 60%, reflecting the trade-off between PREC and REC.

Notably, in the "Barely True" category (class 3), the model achieved a REC of 100%, implying that it correctly identified all instances of this class. However, the PREC for this class was 0%, indicating that when it predicted an article as barely true, it was always incorrect. This result highlights a specific challenge in classifying articles as barely true, where the model struggled to provide accurate predictions.

The model attained an ACC of 62%, demonstrating its ability to classify articles into multiple categories. However, the varying performance across classes suggests potential areas for improvement, particularly in enhancing PREC and REC for certain categories.

	precision	recall	f1-score	support
0	0.64	0.65	0.64	4650
1	0.62	0.64	0.63	4154
2	0.94	0.44	0.60	1120
3	0.00	1.00	0.00	1
accuracy			0.62	9925
macro avg	0.55	0.68	0.47	9925
weighted avg	0.66	0.62	0.63	9925

Figure 4.119. Performance Evaluation of LSTM-AutoEncoder

CNN-GRU

Throughout the training phase of the CNN-GRU model across five distinct folds, we observed a consistent progression in loss and ACC metrics. Each fold comprised five epochs, during which the model underwent fine-tuning using a sequential approach (Figure 4.120). In the first fold, the initial epoch displayed a training ACC of 81.04%, which experienced a significant increase, reaching 95.61% by the end of the fifth epoch. Simultaneously, the validation ACC started promisingly at 94.70% and peaked at 95.35% in the third epoch before experiencing a slight decline in subsequent epochs. Loss metrics exhibited an inverse behaviour compared to ACC, with training loss decreasing from 38.91% to 7.34% in the final epoch. Validation loss, on the other hand, reached its lowest point at 8.52% in the second epoch before increasing slightly. As we progressed through the subsequent folds (2 to 5), a stabilisation pattern became apparent in the ACC curves, oscillating within a narrow range and consistently achieving validation accuracies above the 95% mark. This achievement highlighted the model's ability to generalize its learning from the training data

to unseen data. Notably, the final fold demonstrated the highest performance, concluding with a test ACC of 95.78%. The loss trajectories across all folds initially followed a downward trend, stabilizing as the epochs advanced, with occasional minor increases. Notably, by the end of the fifth fold, the model achieved its lowest validation loss at 6.76%, indicating minimal discrepancies between predicted and actual values and underscoring the model's learning effectiveness. Furthermore, the test losses and accuracies recorded at the end of each fold reaffirmed the model's robustness, with loss values decreasing and ACC increasing, reflecting the model's evolving proficiency over the sequential folds.

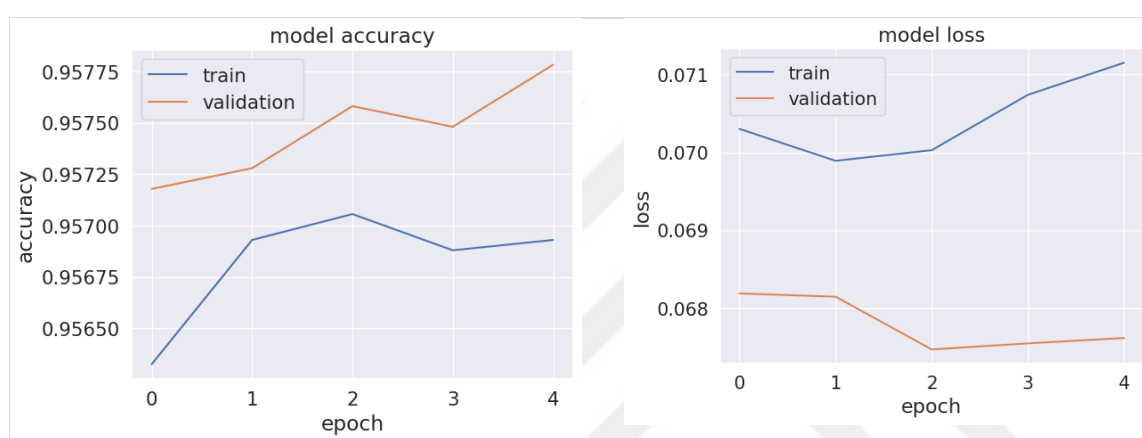


Figure 4.120. Accuracy and loss metrics for training and validation of CNN-GRU

The presented CM for the CNN-GRU model offers a detailed representation of the model's performance in categorizing observations into four distinct categories: fake, true, half-true, and barely true (Figure 4.121). When we examine the matrix's main diagonal, it becomes apparent that the model has exhibited highly accurate predictive capabilities for the "fake" and "true" classifications. It achieved 4694 correct predictions out of 4696 for the former and a perfect classification score of 4283 for the latter. In the case of "half true" and "barely true," the model's performance shows slightly lower but still significant levels of correct predictions, with 524 out of 526 for the former and 5 out of 420 for the latter. However, analysing the off-diagonal elements is essential to gain insight into misclassifications. For example, "fake" has been erroneously classified as "half true" twice, and there is one instance each of "half true" being mistaken for "true" and "barely true." Furthermore, predictions for the "barely true" category appear to have a slightly higher misclassification rate, with 415 instances being categorized as "half true." It's crucial to note that the second and fourth columns of the CM have significantly fewer non-zero entries, indicating a high PREC for "true" and "barely true" classifications, respectively.

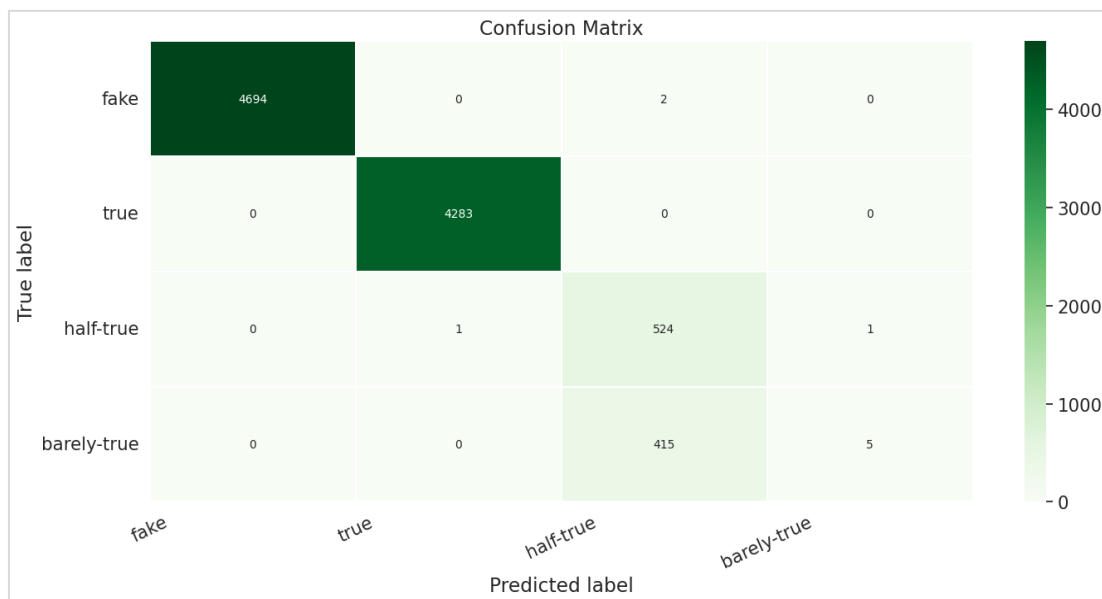


Figure 4.121. Evaluation Matrix for CNN-GRU

The presented CM for the CNN-GRU model offers a detailed representation of the model's performance in categorizing observations into four distinct categories: fake, true, half-true, and barely true (Figure 4.121). When we examine the matrix's main diagonal, it becomes apparent that the model has exhibited highly accurate predictive capabilities for the "fake" and "true" classifications. It achieved 4694 correct predictions out of 4696 for the former and a perfect classification score of 4283 for the latter. In the case of "half true" and "barely true," the model's performance shows slightly lower but still significant levels of correct predictions, with 524 out of 526 for the former and 5 out of 420 for the latter. However, analysing the off-diagonal elements is essential to gain insight into misclassifications. For example, "fake" has been erroneously classified as "half true" twice, and there is one instance each of "half true" being mistaken for "true" and "barely true." Furthermore, predictions for the "barely true" category appear to have a slightly higher misclassification rate, with 415 instances being categorized as "half true." It's crucial to note that the second and fourth columns of the CM have very few non-zero entries, indicating a high PREC for "true" and "barely true" classifications, respectively.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4694
1	1.00	1.00	1.00	4284
2	1.00	0.56	0.71	941
3	0.01	0.83	0.02	6
accuracy			0.96	9925
macro avg	0.75	0.85	0.68	9925
weighted avg	1.00	0.96	0.97	9925

Figure 4.122. Performance Evaluation of CNN-GRU

Discussion

The exploration of DL algorithms on dataset 2 has provided valuable insights into the classification of FND in a multiclass setting (Table 4.8). The LSTM model exhibited a strong overall ACC of 98.64%, achieving perfect ACC in classes 0 and 1. However, there was a noticeable decline in PREC and REC for classes 2 and 3. Specifically, the PREC and REC for the "half-true" category were 93% and 83%, respectively, while for the "barely-true" class, they were 77% and 92%, respectively. Similarly, the CNN-LSTM model achieved an ACC of 96.06% with flawless scores for the "fake" and "true" classes but encountered a significant drop in performance in the "half-true" category and an unexpected result in the "barely-true" class. The PREC and REC for the "half-true" class were 97% and 59%, respectively, while for the "barely-true" category, they were 16% and 68%, respectively. In contrast, the CNN model demonstrated a more modest performance with an ACC of 61.88%. It displayed a more balanced but lower PREC and REC across the four classes, with a strange result for the "barely true" class due to only one support instance. For example, the PREC and REC for the "half-true" class were 94% and 44%, respectively, while for the "barely-true" category, the PREC was 0% despite a 100% REC. Lastly, the CNN-GRU model reported an ACC of 95.77%, once again achieving impeccable results for classes 0 and 1 but experiencing a significant decrease in performance for the newly introduced classes. The PREC and REC for the "half-true" class were 100% and 56%, respectively, while for the "barely-true" category, they were 0.1% and 83%, respectively. These divergent outcomes among the models highlight the varying capabilities of DL techniques in distinguishing between fake, true, half-true, and barely-true news. While the LSTM and CNN-GRU models displayed more consistent and robust performance, the CNN-LSTM and CNN models faced challenges, particularly in handling the more nuanced classes, such as the "barely true" category. Additionally, the LSTM

AutoEncoder model exhibited an ACC of 62%, similar to the CNN model, and also encountered significant difficulties in classifying the "barely-true" category.

Table 4.8. DL Classifiers Results on Dataset2

Classifier	ACC	Class	PREC	REC	F-S
CNN	61.88%	Fake	64%	65%	64%
		Real	62%	64%	63%
		half-true	94%	44%	60%
		barely-true	0%	100%	0%
CNN-LSTM	96.6%	Fake	100%	100%	100%
		Real	100%	100%	100%
		half-true	97%	59%	73%
		barely-true	16%	68%	26%
CNN-GRU	95.77%	Fake	100%	100%	100%
		Real	100%	100%	100%
		half-true	100%	56%	71%
		barely-true	0.1%	83%	0.2%
LSTM	98.64%	Fake	100%	100%	100%
		Real	100%	100%	100%
		half-true	93%	83%	88%
		barely-true	77%	92%	84%
LSTM AutoEncoder	62%	Fake	64%	65%	64%
		Real	62%	64%	63%
		half-true	94%	44%	60%
		barely-true	0%	100%	0%

4.7. Comparing Results

Table 4.9 presents the performance of various ML classifiers across four distinct datasets: ISOT, FakeNewsNet, Dataset1, and Dataset2, revealing varying levels of ACC. The RF classifier delivered solid performance, achieving an ACC of 79% on the ISOT dataset. Notably, its performance soared to 93.85% on the FakeNewsNet dataset, and it maintained respectable accuracies of 93.28% and 88.25% on the Dataset1 and Dataset2, respectively.

GB showcased promising results, with an ACC of 82.24% on the ISOT dataset. It excelled on the FakeNewsNet dataset, achieving an exceptional ACC of 99.54%. Additionally, GB displayed high accuracies of 98.97% and 94.89% on Dataset1 and Dataset2, respectively.

XGBoost emerged as the most consistently high-performing model across all datasets. It achieved an ACC of 81.9% on the ISOT dataset and an astounding ACC of 99.38% on the FakeNewsNet

dataset. XGBoost maintained robustness with accuracies of 99.56% on Dataset 1 and 95.19% on Dataset 2.

Linear SVM exhibited notable variability in performance. It achieved a relatively low ACC of 34.15% on the ISOT dataset but displayed a remarkable turnaround on the FakeNewsNet dataset, reaching an ACC of 99.67%. For Dataset1 and Dataset2, the accuracies were 98.97% and 91.08%, respectively.

RBF SVM demonstrated consistency with a slight dip in performance on the ISOT dataset, scoring 83.57%. The FakeNewsNet dataset achieved an ACC of 99.76%, while for Dataset1 and Dataset2, the accuracies were 92.99% and 94.89%, respectively.

LR performed similarly to RBF SVM on the ISOT dataset, registering an ACC of 74.82%. However, it outperformed RBF SVM on the FakeNewsNet dataset with a 99.04% ACC. LR nearly reached perfection on Dataset 1 with a 98.99% ACC rate and scored 91.75% on Dataset 2.

The DT classifier exhibited an unexpected performance pattern. It achieved an almost perfect ACC of 73.20% on the ISOT dataset but experienced a dip on the FakeNewsNet dataset, with an ACC of only 99.65%. However, it rebounded on Dataset1 and Dataset2, achieving accuracies of 99.56% and 89.73%, respectively.

MNB demonstrated its performance across these datasets, with accuracies of 50.53% on the ISOT dataset, 94.03% on the FakeNewsNet dataset, 89.10% on Dataset1, and 85.01% on Dataset2. While MNB showed relative strength on some datasets, it did not consistently achieve the high ACC levels observed in other classifiers across all datasets, indicating its suitability may vary depending on the dataset and classification task.

Table 4.9. ACC of ML classifiers across datasets

Classifier	ISOT	FakeNewsNet	Dataset1	Dataset2
RF	79%	93.85%	93.28%	88.25%
GB	82.24%	99.54%	98.97%	94.89%
XGBoost	81.9%	99.38%	99.56%	95.19%
Linear SVM	34.15%	99.67%	98.97%	91.08%
RBF SVM	83.57%	99.76%	92.99%	94.89%
LR	74.82%	99.04%	98.99%	91.75%
DT	73.20%	99.65%	99.56%	89.73%
MNB	50.53%	94.03%	89.10%	85.01%

Table 4.10 offers a valuable comparative analysis of the performance of five distinct DL models, CNN, CNN-LSTM, CNN-GRU, LSTM, and LSTM AutoEncoder, across four diverse datasets: ISOT, FakeNewsNet, Dataset1, and Dataset2. This table underscores the significant variability in model performance depending on the specific dataset employed, shedding light on the effectiveness of each model in varying contexts.

When examining the ISOT dataset, the LSTM model emerged as the unequivocal leader, achieving an almost perfect ACC of 99.95%. The CNN model also delivered a commendable performance, boasting an ACC of 98.59%. CNN-LSTM and CNN-GRU closely trailed behind, hovering around the 98% mark, specifically recording ACCs of 98.03% and 97.99%, respectively.

Shifting focus to the FakeNewsNet dataset, the CNN-GRU model took the spotlight with an astonishing ACC of 99.97%. Interestingly, the CNN-LSTM model followed closely as the runner-up, boasting an ACC of 97.99%. The LSTM model performed well but not as impressively, achieving a score of 93.74%, while the CNN model experienced a significant drop in performance, plummeting to a mere 69.07%.

Dataset 1 exhibited remarkable outcomes for the CNN-LSTM and LSTM models, reporting near-perfect accuracies of 99.90%. CNN-GRU was slightly behind, registering an ACC of 99.96%. In contrast, the CNN model underperformed substantially, with an ACC of 59.34%.

In the case of Dataset 2, the LSTM model took the lead with an ACC of 98.64%. The CNN-LSTM closely followed with an ACC of 96.6%, while the CNN-GRU model achieved a solid ACC of 95.77%. Meanwhile, the CNN model's performance remained modest, with an ACC of 61.88%.

Consistently across all these datasets, the LSTM AutoEncoder model stood out as an outlier. Its performance was notably weaker than its counterparts, with ACC ranging between 49% and 62% across the four datasets. This stark contrast underscores the significant disparities in performance that can arise when different models are applied to the same datasets.

Table 4.10. ACC of DL classifiers Across Datasets

Model	ISOT	FakeNewsNet	Dataset1	Dataset2
CNN	98.59%	69.07%	59.34%	61.88%
CNN-LSTM	98.03%	97.99%	99.90%	96.6%
CNN-GRU	98.03%	97.99%	99.96%	95.77%
LSTM	99.95%	93.74%	99.90%	98.64%
LSTM AutoEncoder	97.99%	99.97%	49%	96.6%

4.8. Discussion

Our study distinguishes itself from previous research in the realm of FND (FND) through several key aspects. Firstly, our methodology incorporates a comprehensive and multi-dimensional approach by integrating datasets such as the ISOT corpus, FakeNewsNet, and the LIAR-PLUS dataset. This amalgamation facilitates a nuanced examination, surpassing basic binary classification to include more subtle categories like "Half-True" and "Barely True." This layered and hierarchical assembly of data is designed to progressively enhance the complexity and depth of our classification method, addressing the multifaceted challenges posed by fake news more thoroughly.

Furthermore, our study employs various ML and DL classifiers, ranging from RF and XGBoost to CNN, LSTM, and LSTM AutoEncoder. Including various models allows for a more comprehensive evaluation of performance across different datasets, shedding light on the effectiveness of each model in varying contexts.

In terms of results, our study showcases varying levels of ACC for ML classifiers across datasets such as ISOT, FakeNewsNet, Dataset1, and Dataset2. Notably, the XGBoost model is consistently high-performing across all datasets, achieving remarkable ACC levels. Linear SVM exhibits notable variability in performance, and the RF classifier delivers solid performance with noteworthy ACC on the FakeNewsNet dataset. This contrasts with previous studies that may not have explored such a diverse range of classifiers or datasets, showcasing our study's unique contributions and insights to the field.

In the realm of DL, our study reveals significant variability in the performance of models such as CNN, CNN-LSTM, CNN-GRU, LSTM, and LSTM AutoEncoder across different datasets. The LSTM model, in particular, stands out with near-perfect ACC on the ISOT

dataset. Interestingly, the LSTM AutoEncoder model consistently performs less than its counterparts, highlighting the importance of considering the specific characteristics of datasets in selecting appropriate DL models.

Our study contributes to the growing body of knowledge on FND. It distinguishes itself through integrating diverse datasets and exploring a wide array of classifiers and models. The results demonstrate the nuanced nature of FND, emphasizing the need for a tailored and context-specific approach to addressing the evolving challenges posed by misinformation in the digital landscape.



5. CONCLUSION AND FUTURE WORK

5.1. Conclusion

In summary, our thorough investigation provides a fundamental investigation into the potential of several ML and DL models for identifying false information in various applications. To enable a thorough evaluation of every model's performance, we made use of several datasets, each with distinct properties. Traditional machine learning algorithms like RF, Linear SVM, and DT were evaluated alongside sophisticated deep learning frameworks like CNN, LSTM, and hybrid versions. Our analysis underscored the critical role of the dataset and classification task in determining the efficacy of a model. While certain models consistently achieved high ACC across all datasets, others demonstrated exceptional performance in specific scenarios. This highlights the absence of a universal solution for addressing the intricate challenge of FND. Instead, our results highlight the necessity for a multimodal strategy, where optimal results can be achieved by combining models that are specifically adapted to the particulars of the task at hand. With the support of this strong technique, we can better protect the truth and maintain the integrity of information sharing in a world that is becoming increasingly digital.

5.2. Future Work

The landscape of research in this rapidly evolving field offers many exciting opportunities. Firstly, as technology continues to advance, the methods employed in creating and disseminating fake news are likely to become increasingly sophisticated. We must maintain a continuous process of adaptation and enhancement of our models to stay ahead of these evolving tactics. Another intriguing research direction is incorporating elements like sentiment analysis or language patterns frequently observed in false news. These additions may improve the predictive ACC of our models. The creation of ensemble approaches, which combine the advantages of several models, offers a viable plan for constructing a stronger barrier against false information.

Furthermore, the need for more diverse and representative datasets cannot be overstated. Such datasets should encompass various languages, regional contexts, and media formats to ensure the universality and applicability of our models. By pursuing these future directions,

our objective is continuously refine our understanding and capabilities. This ongoing effort is vital to ensure our models remain effective in the ongoing battle against misinformation.



REFERENCES

1. İnternet: Data Reportal. (2023). *Social Media Users*. URL: <https://datareportal.com/social-media-users>. Son Erişim Tarihi: 17.05.2023.
2. İnternet: Chaffey, D. (2023). *Global social media statistics research summary 2023*. URL: <https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/> Son Erişim Tarihi: 17.05.2023.
3. Aslam, N., Ullah Khan, I., Alotaibi, F. S., Aldaej, L. A., and Aldubaikil, A. K. (2021). Fake detect: A deep learning ensemble model for fake news detection. *Complexity*, 1, 1-8.
4. Adewole, K. S., Han, T., Wu, W., Song, H., and Sangaiah, A. K. (2020). Twitter spam account detection based on clustering and classification methods. *The Journal of Supercomputing*, 76, 4802-4837.
5. Mishra, S., Shukla, P., and Agarwal, R. (2022). Analyzing machine learning enabled fake news detection techniques for diversified datasets. *Wireless Communications and Mobile Computing*, 1, 1-18.
6. Ling, R. (2020). Confirmation bias in the era of mobile news consumption: the social and psychological dimensions. *Digital Journalism*, 8(5), 596-604.
7. Pennycook, G., and Rand, D. G. (2021). The psychology of fake news. *Trends in Cognitive Sciences*, 25(5), 388-402.
8. Zhang, X., and Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, 57(2), 102025.
9. Tandoc, Jr., E. C., Lim, Z. W., and Ling, R. (2018). Defining “fake news” A typology of scholarly definitions. *Digital Journalism*, 6(2), 137-153.
10. Wardle, C., and Derakhshan, H. (2018). Thinking about ‘information disorder’: formats of misinformation, disinformation, and mal-information. *Journalism, ‘Fake News’ & Disinformation*, 1, 43-54.
11. Antinori, A. (2019, October). *Terrorism and deepfake: From hybrid warfare to post-truth warfare in a hybrid world*. In ECIAIR 2019 European Conference on the Impact of Artificial Intelligence and Robotics. Academic Conferences and Publishing Limited, 23.
12. Duffy, A., Tandoc, E., and Ling, R. (2020). Too good to be true, too good not to share: the social utility of fake news. *Information, Communication & Society*, 23(13), 1965-1979.
13. Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1), 22-36.

14. El Rayess, M., Chebl, C., Mhanna, J., and Hage, R. M. (2018). Fake news judgement: The case of undergraduate students at Notre Dame University-Louaize, Lebanon. *Reference Services Review*, 46(1), 146-149.
15. Dalkir, K., and Katz, R. (Eds.). (2020). *Navigating fake news, alternative facts, and misinformation in a post-truth world*. IGI Global.
16. Dubey, S., Biswas, P., Ghosh, R., Chatterjee, S., Dubey, M. J., Chatterjee, S., and Lavie, C. J. (2020). Psychosocial impact of COVID-19. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 14(5), 779-788.
17. Fleming, N. (2020). Coronavirus misinformation, and how scientists can help to fight it. *Nature*, 583(7814), 155-156.
18. Molina, M. D., Sundar, S. S., Le, T., and Lee, D. (2021). "Fake news" is not simply false information: A concept explication and taxonomy of online content. *American Behavioral Scientist*, 65(2), 180-212.
19. Mourão, R. R., and Robertson, C. T. (2019). Fake news as discursive integration: An analysis of sites that publish false, misleading, hyperpartisan and sensational information. *Journalism Studies*, 20(14), 2077-2095.
20. Johansson, P., Enock, F., Hale, S., Vidgen, B., Bereskin, C., Margetts, H., and Bright, J. (2022). How can we combat online misinformation? A systematic overview of current interventions and their efficacy. *arXiv preprint arXiv:2212.11864*.
21. Nistor, A., and Zadobrischi, E. (2022). The influence of fake news on social media: analysis and verification of web content during the COVID-19 pandemic by advanced machine learning methods and natural language processing. *Sustainability*, 14(17), 10466.
22. Choraś, M., Demestichas, K., Giełczyk, A., Herrero, Á., Ksieniewicz, P., Remoundou, K., and Woźniak, M. (2021). Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study. *Applied Soft Computing*, 101, 107050.
23. Tang, L., and Mahmoud, Q. H. (2021). A deep learning-based framework for phishing website detection. *Access*, 10, 1509-1521.
24. Frank, R. (2015). Caveat lector: Fake news as folklore. *Journal of American Folklore*, 128(509), 315-332.
25. Uppada, S. K., and Patel, P. (2023). An image and text-based multimodal model for detecting fake news in osn's. *Journal of Intelligent Information Systems*, 61(2), 367-393.
26. Kumar, K. K., and Geethakumari, G. (2014). Detecting misinformation in online social networks using cognitive psychology. *Human-Centric Computing and Information Sciences*, 4(1), 14.

27. Wang, X., and Song, Y. (2020). Viral misinformation and echo chambers: The diffusion of rumors about genetically modified organisms on social media. *Internet Research*, 30(5), 1547-1564.
28. Parikh, S. B., and Atrey, P. K. (2018, April). Media-rich fake news detection: A survey. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*. State University of New York, Albany, NY, USA, 436-441.
29. Vukasović, T., and Strašek, R. (2014). A study on effective communication strategy in developing brand communication: analysis of social networking site. In *Proceedings of the World Congress on Engineering*. London, U.K., 690-693.
30. Haenlein, M., Anadol, E., Farnsworth, T., Hugo, H., Hunichen, J., and Welte, D. (2020). Navigating the New Era of Influencer Marketing: How to be Successful on Instagram, TikTok, & Co. *California Management Review*, 63(1), 5-25.
31. Williams, A. P., Trammell, K. D., Postelnicu, M., Landreville, K. D., and Martin, J. D. (2005). Blogging and hyperlinking: Use of the Web to enhance viability during the 2004 US campaign. *Journalism Studies*, 6(2), 177-186.
32. Vasilakes, J., Zhou, S. and Zhang, R. (2020). Chapter 6 - Natural language processing. *Machine Learning in Cardiovascular Medicine*, 1(4), 123–148.
33. Tao, D., Yang, P., and Feng, H. (2020). Utilization of text mining as a big data analysis tool for food science and nutrition. *Comprehensive Reviews in Food Science and Food Safety*, 19(2), 875-894.
34. Fattahi, J., and Mejri, M. (2021, January). SpaML: a bimodal EL spam detector based on NLP techniques. In *2021 IEEE 5th international conference on cryptography, security and privacy (CSP)*. Zhuhai, China, 107-112.
35. Kastrati, Z., Dalipi, F., Imran, A. S., Pireva Nuci, K., and Wani, M. A. (2021). Sentiment analysis of students' feedback with NLP and deep learning: A systematic mapping study. *Applied Sciences*, 11(9), 3986.
36. Khan, N. S., Abid, A., and Abid, K. (2020). A novel natural language processing (NLP)-based machine translation model for English to Pakistan sign language translation. *Cognitive Computation*, 12, 748-765.
37. Yan, H., Deng, B., Li, X., and Qiu, X. (2019). TENER: adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474*.
38. Adhikari, S. (2020, March). Nlp based machine learning approaches for text summarization. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)* (). Erode, India, 535-538.
39. Adam, D. E. (2020). Deep learning based NLP techniques in text to speech synthesis for communication recognition. *Journal of Soft Computing Paradigm*, 2(4), 209-215.

40. López-Úbeda, P., Martín-Noguerol, T., Aneiros-Fernández, J., and Luna, A. (2022). Natural language processing in pathology: current trends and future insights. *The American Journal of Pathology*, 192(11), 1486–1495.
41. Kumar, A., and Sarmiento, P. (2023). From Words to Music: A Study of Subword Tokenization Techniques in Symbolic Music Generation. *arXiv preprint arXiv:2304.08953*.
42. Tunstall, L., Von Werra, L., and Wolf, T. (2022). *Natural Language Processing with Transformers, Revised Edition*. O'Reilly Media, Incorporated.
43. İnternet: Shahid, A. (2023). *Natural Language Processing Techniques for Sentiment Analysis in Social Media*, Prepr. Ur: <https://accounts.osf.io/login?service=https://osf.io/qgdbz/download> Son Erişim Tarihi: 17.05.2023.
44. İnternet: Rahi, S., and Tahir, Z. (2022). Assessing factors which influence individual behaviour to adopt family takaful. *Economic and Social Development: Book of Proceedings*, 1-5. URL: https://www.bib.irb.hr/1175551/download/1175551.Book_of_Proceedings_esdJanuary2022_Online.pdf#page=6 Son Erişim Tarihi: 17.05.2023.
45. Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5-32.
46. Moshkov, M. (2022). Decision trees for regular factorial languages. *Array*, 15, 100203.
47. Chiu, M. H., Yu, Y. R., Liaw, H. L., and Chun-Hao, L. (2016). The use of facial micro-expression state and Tree-Forest Model for predicting conceptual-conflict based conceptual change. *Chapter Title & Authors*, 1, 184-192.
48. Biresselioglu, M. E., and Demir, M. H. (2022). Constructing a Decision Tree for Energy Policy Domain Based on Real-Life Data. *Energies*, 15(7), 2420.
49. Minsky, M. (1961). Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1), 8-30.
50. [50] Noble, W. S. (2006). What is a support vector machine?. *Nature Biotechnology*, 24(12), 1565-1567.
51. Ghosh, S., Dasgupta, A., and Swetapadma, A. (2019, February). A study on support vector machine based linear and non-linear pattern classification. In *2019 International Conference on Intelligent Sustainable Systems (ICISS)*. Palladam, India, 24-28.
52. [Almaiah, M. A., Almomani, O., Alsaaidah, A., Al-Otaibi, S., Bani-Hani, N., Hwaitat, A. K. A., and Aldhyani, T. H. (2022). Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels. *Electronics*, 11(21), 3571.
53. Kumar, A., Beniwal, R., and Jain, D. (2023). Personality detection using kernel-based ensemble model for leveraging social psychology in online networks. *ACM*

Transactions on Asian and Low-Resource Language Information Processing, 22(5), 1-20.

54. Chamasemani, F. F., and Singh, Y. P. (2011, September). *Multi-class support vector machine (SVM) classifiers--an application in hypothyroid detection and classification*. In 2011 sixth international conference on bio-inspired computing: theories and applications. Penang, Malezya, 351-356.
55. Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367-378.
56. Ho, W. K., Tang, B. S., and Wong, S. W. (2021). Predicting property prices with machine learning algorithms. *Journal of Property Research*, 38(1), 48-70.
57. Johnson, J. M., and Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 1-54.
58. Bordes, A., Glorot, X., Weston, J., and Bengio, Y. (2012, March). *Joint learning of words and meaning representations for open-text semantic parsing*. In Artificial intelligence and statistics. PMLR, 127-135.
59. Cireşan, D. C., Meier, U., and Schmidhuber, J. (2012, June). Transfer learning for Latin and Chinese characters with deep neural networks. In *The 2012 international joint conference on neural networks (IJCNN)*. Brisbane, QLD, Australia, 1-6.
60. Vani, R. (2017). Towards efficient intrusion detection using deep learning techniques: a review. *International Journal of Advanced Research in Computer and Communication Engineering*, 6(10), 375-384.
61. Guo, Y., Cao, X., Liu, B., and Peng, K. (2020). El Niño index prediction using deep learning with ensemble empirical mode decomposition. *Symmetry*, 12(6), 893.
62. Bibi, I., Akhunzada, A., Malik, J., Iqbal, J., Musaddiq, A., and Kim, S. (2020). A dynamic DL-driven architecture to combat sophisticated Android malware. *IEEE Access*, 8, 129600-129612.
63. Lin, Z., Lai, J., Chen, X., Cao, L., and Wang, J. (2022). Curriculum reinforcement learning based on K-fold cross validation. *Entropy*, 24(12), 1787.
64. Widodo, S., Brawijaya, H., and Samudi, S. (2022). Stratified K-fold cross validation optimization on machine learning for prediction. *Sinkron: jurnal dan penelitian teknik informatika*, 7(4), 2407-2414.
65. Prusty, S., Patnaik, S., and Dash, S. K. (2022). SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. *Frontiers in Nanotechnology*, 4, 972421.
66. İnternet: Maredia, R. (2020). Analysis of Google Play Store Data set and predict the popularity of an app on Google Play Store. Url: https://www.researchgate.net/publication/343769728_Analysis_of_Google_Play_Store_Data_set_and_predict_the_popularity_of_an_app_on_Google_Play_Store [Accessed 24th December 2021].

67. Ahmad, I., Yousaf, M., Yousaf, S., and Ahmad, M. O. (2020). Fake news detection using machine learning ensemble methods. *Complexity*, 1, 1-11.
68. Khan, J. Y., Khondaker, M. T. I., Afroz, S., Uddin, G., and Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, 4, 100032.
69. Baarir, N. F., and Djeffal, A. (2021, February). Fake news detection using machine learning. In *2020 2nd International workshop on human-centric smart environments for health and well-being (IHSH)*. Boumerdes, Algeria, 125-130.
70. Shaikh, J., and Patil, R. (2020, December). Fake news detection using machine learning. In *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*. Gunupur Odisha, India, 1-5.
71. Pérez-Rosas, V., Kleinberg, B., Lefevre, A., and Mihalcea, R. (2017). Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.
72. Braşoveanu, A. M., and Andonie, R. (2021). Integrating machine learning techniques in semantic fake news detection. *Neural Processing Letters*, 53(5), 3055-3072.
73. Chiramel, S., Logofătu, D., and Goldenthal, G. (2020, October). Detection of social media platform insults using Natural language processing and comparative study of machine learning algorithms. In *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*. Sinaia, Romania, 98-101.
74. Hiramath, C. K., and Deshpande, G. C. (2019, July). Fake news detection using deep learning techniques. In *2019 1st International Conference on Advances in Information Technology (ICAIT)*. Chikmagalur, India, 411-415.
75. Jiang, T. A. O., Li, J. P., Haq, A. U., Saboor, A., and Ali, A. (2021). A novel stacking approach for accurate detection of fake news. *Access*, 9, 22626-22639.
76. Reddy, H., Raj, N., Gala, M., and Basava, A. (2020). Text-mining-based fake news detection using ensemble methods. *International Journal of Automation and Computing*, 17(2), 210-221.
77. Ozbay, F. A., and Alatas, B. (2020). Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: Statistical Mechanics and its Applications*, 540, 123174.
78. Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2020). Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, 8(3), 171-188.
79. Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1), 22-36.
80. Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1), 22-36.

81. Alhindi, T., Petridis, S., and Muresan, S. (2018, November). *Where is your evidence: Improving fact-checking by justification modeling*. In Proceedings of the first workshop on fact extraction and verification (FEVER). Brussels, Belgium, 85-90.
82. Fauzi, A., Setiawan, E. B., and Baizal, Z. K. A. (2019, March). Hoax news detection on Twitter using term frequency inverse document frequency and support vector machine method. In *Journal of Physics: Conference Series* (012025). China: IOP Publishing.
83. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., and Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14, 1-37.
84. Ho, T. K. (1995, August). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (278-282). Montreal, QC, Canada.
85. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1, 1189-1232.
86. Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 20(2), 215-232.
87. Chen, T., and Guestrin, C. (2016, August). *Xgboost: A scalable tree boosting system*. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (785-794). San Francisco.
88. Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20, 273-297.
89. Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
90. Raza, S., and Ding, C. (2022). Fake news detection based on news content and social contexts: a transformer-based approach. *International Journal of Data Science and Analytics*, 13(4), 335-362.
91. Hangloo, S., and Arora, B. (2021). Fake News Detection Tools and Methods--A Review. *arXiv preprint arXiv:2112.11185*.
92. Han, Y., Silva, A., Luo, L., Karunasekera, S., and Leckie, C. (2021). Knowledge enhanced multi-modal fake news detection. *arXiv preprint arXiv:2108.04418*.
93. Wang, X., Chao, F., Ma, N., and Yu, G. (2022). Exploring the effect of spreading fake news debunking based on social relationship networks. *Frontiers in Physics*, 10, 833385.
94. Mazzeo, V., and Rapisarda, A. (2022). Investigating fake and reliable news sources using complex networks analysis. *Frontiers in Physics*, 10, 886544.

95. Kim, B., Xiong, A., Lee, D., and Han, K. (2021). A systematic review on fake news research through the lens of news creation and consumption: Research efforts, challenges, and future directions. *PloS one*, 16(12), e0260080.
96. Chiche, A., and Yitagesu, B. (2022). Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1), 1-25.
97. Heid, S., Wever, M., and Hüllermeier, E. (2020). Reliable part-of-speech tagging of historical corpora through set-valued prediction. *arXiv preprint arXiv:2008.01377*.
98. Janiesch, C., Zschech, P., and Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3), 685-695.
99. Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons b*, 4, 51-62.
100. James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J. (2023). *Unsupervised learning*. In *An Introduction to Statistical Learning: with Applications in Python* (pp. 503-556). Cham: Springer International Publishing.
101. Van Engelen, J. E., and Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2), 373-440.
102. Dayan, P., and Niv, Y. (2008). Reinforcement learning: the good, the bad and the ugly. *Current Opinion in Neurobiology*, 18(2), 185-196.
103. [103] Walczak, S. (2019). Artificial neural networks. In *Advanced methodologies and technologies in artificial intelligence, computer simulation, and human-computer interaction* (pp. 40-53). IGI global.
104. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., and Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354-377.
105. Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., and Ghayvat, H. (2021). CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, 10(20), 2470.
106. Ma, W., & Lu, J. (2017). An equivalence of fully connected layer and convolutional layer. *arXiv preprint arXiv:1712.01252*.
107. Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017, August). *Understanding of a convolutional neural network*. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Antalya, Türkiye.
108. Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Science*, 6(12), 310-316.
109. Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

110. Mishra, S., Sachan, R., and Rajpal, D. (2020). Deep convolutional neural network based detection system for real-time corn plant disease recognition. *Procedia Computer Science*, 167, 2003-2010.
111. Basha, S. S., Dubey, S. R., Pulabaigari, V., and Mukherjee, S. (2020). Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378, 112-119.
112. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.
113. Internet: Baldi, P., and Sadowski, P. J. (2013). Understanding dropout. *Advances in Neural Information Processing Systems*, 26. URL: <https://proceedings.neurips.cc/paper/2013/hash/71f6278d140af599e06ad9bf1ba03cb0-Abstract.html> Son Erişim Tarihi: 17.05.2023.
114. Internet: Bjorck, N., Gomes, C. P., Selman, B., and Weinberger, K. Q. (2018). *Understanding batch normalization*. *Advances in Neural Information Processing Systems*, 31. URL: <https://proceedings.neurips.cc/paper/2018/hash/36072923bfc3cf47745d704feb489480-Abstract.html> Son Erişim Tarihi: 17.05.2023.
115. Grossi, E., and Buscema, M. (2007). Introduction to artificial neural networks. *European Journal of Gastroenterology & Hepatology*, 19(12), 1046-1054.
116. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., and Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*.
117. Nosouhian, S., Nosouhian, F., and Khoshouei, A. K. (2021). A review of recurrent neural network architecture for sequence learning: Comparison between LSTM and GRU. *Preprints 2021*, 2021070252.
118. Graves, A., and Graves, A. (2012). Long short-term memory. *Supervised Sequence Labelling With Recurrent Neural Networks*, 1, 37-45.
119. Van Houdt, G., Mosquera, C., and Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, 53, 5929-5955.
120. Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7), 1235-1270.
121. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
122. Sachin, S., Tripathi, A., Mahajan, N., Aggarwal, S., and Nagrath, P. (2020). Sentiment analysis using gated recurrent neural networks. *SN Computer Science*, 1, 1-13.

123. Aloysius, N., and Geetha, M. (2017, April). A review on deep convolutional neural networks. In *2017 international conference on communication and signal processing (ICCSP)*. Chennai, India, 0588-0592.
124. Hossin, M., and Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2), 1.
125. Chiu, M. H., Yu, Y. R., Liaw, H. L., and Chun-Hao, L. (2016). The use of facial micro-expression state and Tree-Forest Model for predicting conceptual-conflict based conceptual change. *Chapter Title & Authors*, 1, 1-5.
126. İnternet: Naïve bayes classifier from scratch with hands on examples in r. (2020, April 04). URL: <https://insightimi.wordpress.com/2020/04/04/naive-bayes-classifier-from-scratch-with-hands-on-examples-in-r> Son Erişim Tarihi: 17.05.2023.
127. Fu, Y., Yin, Z., Su, M., Wu, Y., and Liu, G. (2020). Construction and reasoning approach of belief rule-base for classification base on decision tree. *Access*, 8, 138046-138057.
128. Johnson, N., and Rajadurai, K. (2020). Impact of community radio programs in rural development. *International Journal of Scientific & Technology Research*, 9(1), 3844-3848.
129. Alexander, T., Mazurczyk, W., Mishra, A., and Perotti, A. (2019). Mobile communications and networks. *Communications Magazine*, 57(4), 94-94.
130. [130] Ferreira, M., Lopes, B., Granado, A., Freitas, H., and Loureiro, J. (2021). Audio-visual tools in science communication: the video abstract in ecology and environmental sciences. *Frontiers in Communication*, 6, 596248.
131. Tsai, C.-F., and Hung, C. (2013). Popular research topics in multimedia. *Scientometrics*, 95(1), 465–479. doi: 10.1007/s11192-012-0932-1
132. Brennan, E. B. (2021). Using Video to Give More Effective and Engaging Science Talks. *Frontiers in Communication*, 6, 725811.
133. Borah, P. (2014). The hyperlinked world: A look at how the interactions of news frames and hyperlinks influence news credibility and willingness to seek information. *Journal of Computer-Mediated Communication*, 19(3), 576-590.
134. Baggio, R., and Corigliano, M. A. (2009). *On the Importance of Hyperlinks: A Network Science Approach*. Information and Communication Technologies in Tourism 2009. Springer, Vienna. doi: 10.1007/978-3-211-93971-0_26
135. İnternet: Fürsich, E. (2023, October 04). *Textual Analysis and Communication*. URL: <https://www.oxfordbibliographies.com/display/document/obo-9780199756841/obo-9780199756841-0216.xml> Son Erişim Tarihi: 17.05.2023.
136. Kessler, S. H., and Schäfer, M. S. (2022). *Content analysis in the research field of science coverage. standardisierte inhaltsanalyse in der kommunikationswissenschaft – standardized content analysis in communication research*. Springer VS.

137. Rai, A., and Borah, S. (2021). Study of various methods for tokenization. In *Applications of Internet of Things: Proceedings of ICCCIOT 2020* (pp. 193-200). Springer Singapore. doi: 10.1007/978-981-15-6198-6_18
138. Treviso, M., Lee, J. U., Ji, T., Aken, B. V., Cao, Q., Ciosici, M. R., and Schwartz, R. (2023). Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 11, 826-860.
139. Song, X., Salcianu, A., Song, Y., Dopson, D., and Zhou, D. (2020). Fast wordpiece tokenization. *arXiv preprint arXiv:2012.15524*.
140. de Oliveira, N. R., Pisa, P. S., Lopez, M. A., de Medeiros, D. S. V., and Mattos, D. M. (2021). Identifying fake news on social networks based on natural language processing: trends and challenges. *Information*, 12(1), 38.
141. Pramana, R., Subroto, J. J., and Gunawan, A. A. S. (2022, November). *Systematic literature review of stemming and lemmatization performance for sentence similarity*. In 2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA) (1-6). Yogyakarta, Indonesia.
142. Samir, A., and Lahbib, Z. (2018). *Stemming and lemmatization for information retrieval systems in amazigh language*. In Big Data, Cloud and Applications: Third International Conference, BDCA 2018, Kenitra, Morocco, April 4–5, 2018, Revised Selected Papers 3 (pp. 222-233). Springer International Publishing.
143. Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3), 160.
144. Katal, A., and Singh, N. (2022). Artificial neural network: models, applications, and challenges. *Innovative Trends in Computational Intelligence*, 1, 235-257.
145. Kibriya, A. M., Frank, E., Pfahringer, B., and Holmes, G. (2005). *Multinomial naive bayes for text categorization revisited*. In AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17 (488-499). Springer Berlin Heidelberg.
146. Riego, N. C. R., and Villarba, D. B. (2023). Utilization of multinomial naive bayes algorithm and term frequency inverse document frequency (TF-IDF Vectorizer) in checking the credibility of news tweet in the Philippines. *arXiv preprint arXiv:2306.00018*.
147. Staudemeyer, R. C., and Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.
148. Vennerød, C. B., Kjærran, A., and Bugge, E. S. (2021). Long short-term memory RNN. *arXiv preprint arXiv:2105.06756*.
149. Kader, N. I. A., Yusof, U. K., Khalid, M. N. A., and Husain, N. R. N. (2022, September). *A review of long short-term memory approach for time series analysis*

- and forecasting*. In International Conference on Emerging Technologies and Intelligent Systems. Cham: Springer International Publishing, 12-21.
150. Fu, L. (2020, August). *Time series-oriented load prediction using deep peephole lstm*. In 2020 12th International Conference on Advanced Computational Intelligence (ICACI). Dali, China, 86-91.
 151. Michelucci, U. (2022). An introduction to autoencoders. *arXiv preprint arXiv:2201.03898*.
 152. Zhai, J., Zhang, S., Chen, J., and He, Q. (2018, October). *Autoencoder and its various variants*. In 2018 IEEE international conference on systems, man, and cybernetics (SMC). Miyazaki, Japonya, 415-419.
 153. Kingma, D. P., and Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4), 307-392.



Gazili olmak ayrıcalıktır